

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Prise en compte des conditions d'application des normes juridiques dans le cadre d'un système d'aide à la décision judiciaire

Léonard, Sophie

Award date:
1986

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix à Namur
Institut d'Informatique

PRISE EN COMPTE DES CONDITIONS
D'APPLICATION DES NORMES JURIDIQUES
DANS LE CADRE D'UN SYSTEME D'AIDE
A LA DECISION JUDICIAIRE

Mémoire présenté par
Sophie Léonard en
vue de l'obtention
du titre de Licencié
et Maître en
Informatique.

Année Académique 1985-1986

Je tiens beaucoup à remercier ici toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

J'adresse toute ma gratitude à Monsieur le Professeur François Bodart, promoteur, pour m'avoir indiqué les pistes de recherche, pour ses conseils et ses critiques constructives, et à Madame Pascale Stenne, pour la disponibilité, l'encadrement efficace et les encouragements qu'elle m'a prodigués durant toute cette année.

Que le Professeur Jerzy Wroblewski soit également remercié pour avoir accepté de m'accueillir à l'Université de Lodz, et m'avoir donné les éléments indispensables à ce travail.

Je désire aussi exprimer mes remerciements à Monsieur Yves Pouillet, Madame Michèle Hella et Monsieur Marc Schauss, pour m'avoir fourni les apports nécessaires aux niveaux juridique et logique.

Je remercie la société B.I.M. à Everberg, et particulièrement Monsieur Raf Venken, de m'avoir permis d'approfondir mes connaissances en langage Prolog.

Si ce séjour en Pologne fut pour moi aussi passionnant, c'est grâce à la formidable générosité de personnes que je désire citer ici : Le Professeur, Madame et Mademoiselle Rey, Krystina Wojcik, Anna et Sławek, Barbara et Sbešek, Zośia, le Professeur et Madame Andrzej Trybulec, Anna, Lydia et Darek, Beata et Tomek, Iurek.

Je remercie ma famille de m'avoir encouragée tout au long de mes études, ainsi que Christian pour sa gentillesse et sa compréhension.

TABLE DES MATIERES.

TABLE DES MATIERES.	1
INTRODUCTION.	8
PREMIERE PARTIE : ENVIRONNEMENT DU MEMOIRE : PROTOTYPE D'AIDE A L'ELABORATION DE JUGEMENTS	11
CHAPITRE 1 : Présentation d'une synthèse du prototype actuel	12
Section 1.1 : Objectifs du système	12
Section 1.2 : Solution conceptuelle.	13
1.2.1 Les éléments d'un jugement	13
1.2.2 La formalisation du raisonnement du juge	14
1.2.3 Problème juridique étudié.	15
1.2.4 Le schéma conceptuel	16
Section 1.3 : La solution fonctionnelle et sa mise en oeuvre.	22
1.3.1 Fonctions documentaires.	22
1.3.2 Fonctions d'évaluation	23
1.3.2.1 Cohérence interne.	23
1.3.2.1.1 Test de polarisation	24
1.3.2.1.2 Test de pondération.	24
1.3.2.2 Cohérence externe.	24
1.3.3 Implémentation	25
Section 1.4 : Limites du prototype actuel.	25
1.4.1 Limites du modèle sémantique	25
1.4.2 Limites au contrôle de cohérence et de validité.	26

1.4.3 Limites de l'implémentation.	26
CHAPITRE 2 : Extensions envisagées	26
Section 2.1 : Extension des requêtes	27
Section 2.2 : Extension des tests.	28
2.2.1 Tests inter-compositions	29
2.2.1.1 Test des états	29
2.2.1.2 Test des relations	29
2.2.2 Test inter-jugements	30
Section 2.3 : Interface fonctions documentaires - Test de validité externe	31
Section 2.4 : Extension du modèle de raisonnement juridique.	32
Section 2.5 : Etude des formalismes de représentation des concepts utilisés.	33
CHAPITRE 3 : Décisions d'extensions immédiates	34
Section 3.1 : Elargissement du modèle du processus de décision et d'argumentation juridique.	35
Section 3.2 : Analyse des formalismes de représentation des connaissances.	36
DEUXIEME PARTIE : MODELES DES PROCESSUS DE DECISION JUDICIAIRE ET D'ARGUMENTATION JURIDIQUE	37
CHAPITRE 4 : Modèle théorique de la décision juridique proposé par le Professeur J. Wroblewski	38
Section 4.1 : Motivation de la décision judiciaire	38
4.1.1 La décision judiciaire	38
4.1.2 La justification de la décision judiciaire	39
4.1.2.1 Quand intervient la justification ?	39

4.1.2.2	Types de justifications.	40
4.1.2.3	La prise de décision judiciaire.	40
4.1.2.4	Autres significations.	41
Section 4.2 : Décision de validité		42
4.2.1	Concepts de validité	42
4.2.1.1	Validité systémique.	42
4.2.1.2	Validité factuelle	42
4.2.1.3	Validité axiologique	43
4.2.2	Types de systèmes légaux et leurs concepts correspondants de validité	43
4.2.3	Formule.	44
Section 4.3 : Décision d'interprétation.		44
4.3.1	Signification du mot interprétation.	44
4.3.2	Directives d'interprétation.	45
4.3.3	Modèle théorique de l'interprétation légale.	46
4.3.4	Formule.	47
4.3.5	Evaluations.	47
4.3.6	Formalisation du processus d'interprétation.	48
4.3.7	Modèle empirique de l'interprétation	49
4.3.8	Le rôle des principes du droit dans la théorie et l'idéologie de l'interprétation juridique.	49
4.3.8.1	Les types de principes du droit.	49
4.3.8.2	Le rôle des principes dans la théorie de l'interprétation	50
4.3.8.3	Le rôle des principes dans l'idéologie de l'interprétation	51
Section 4.4 : Décision de preuve		52
4.4.1	Que prouver ?	52
4.4.2	Typologie des faits.	52
4.4.2.1	Faits descriptifs ou évaluatifs.	52
4.4.2.2	Faits positifs ou négatifs	53

4.4.2.3 Faits simples ou relationnels.	53
4.4.3 Les directives de preuve	54
4.4.4 Evaluations.	55
4.4.5 Formule.	56
4.4.6 Contrôle de la conviction intime du juge	56
4.4.6.1 Par la loi	56
4.4.6.2 Par la jurisprudence	56
Section 4.5 : L'informatique comme aide au processus judiciaire	57
4.5.1 Modèle du système d'information du processus judiciaire	57
4.5.2 Utilisation de l'informatique dans la prise de décision judiciaire.	59
4.5.2.1 Décision de validité	59
4.5.2.2 Décision d'interprétation.	60
4.5.2.3 Décision de preuve	60
4.5.2.4 Décision de choix de conséquences.	61
4.5.3 Conclusions.	61
Section 4.6 : Avantages et limites du modèle théorique . .	61
Section 4.7 : Extension du système d'aide à la décision judiciaire en fonction de la théorie de la décision juridique	62
4.7.1 Introduction du temps et de l'espace	63
4.7.2 Typologie des faits.	64
CHAPITRE 5 : Modèle empirique de l'argumentation juridique .	65
Section 5.1 : But poursuivi.	65
Section 5.2 : Dispositif d'une décision judiciaire	66
Section 5.3 : Contexte d'application d'un standard	67
Section 5.4 : Méta-règles de l'argumentation basée sur la jurisprudence.	69

Section 5.5 : Avantages et limites du modèle empirique . . .	70
Section 5.6 : Utilisation du modèle empirique dans le SIAD	72
CHAPITRE 6 : Comparaison des modèles théorique et empirique.	73
Section 6.1 : Justification interne d'un jugement.	73
Section 6.2 : Choix du modèle empirique.	75
CHAPITRE 7 : Extension du modèle d'aide à la décision judiciaire : Modèle d'aide à l'argumentation. .	76
Section 7.1 : Objectif du système.	76
Section 7.2 : Définition des concepts.	77
Section 7.3 : Schéma conceptuel des données.	79
CHAPITRE 8 : Fonctions d'aide à l'argumentation.	86
Section 8.1 : Spécification des fonctions.	86
8.1.1 Fonctions d'aide à l'argumentation	86
8.1.2 Fonctions de consultation.	88
8.1.3 Fonctions de mise à jour	88
Section 8.2 : Implémentation des fonctions	91
TROISIEME PARTIE : PERSPECTIVES D'EVOLUTION.	93
CHAPITRE 9 : Critique de la représentation conceptuelle des données.	97
CHAPITRE 10 : Formalismes de représentation des connaissances	102
Section 10.1 : Représentation des connaissances	102
Section 10.2 : Le calcul des prédicats.	104
10.2.1. Définition du calcul des prédicats.	104

10.2.1.1.	Eléments du modèle. Définitions	104
10.2.1.2.	Inférences.	105
10.2.1.3.	Critique générale	106
10.2.2.	Application du calcul des prédicats au modèle juridique	107
10.2.2.1.	Schéma conceptuel	107
10.2.2.2.	Conclusion.	108
10.2.3.	Comparaison du calcul des prédicats avec le schéma Entité/Association.	109
Section 10.3 : Les règles de production		111
10.3.1.	Définition des règles de production	111
10.3.1.1.	Définitions	111
10.3.1.2.	Inférences.	112
10.3.1.3.	Critique générale	112
10.3.2.	Application des règles de production au modèle juridique	113
10.3.2.1.	Schéma conceptuel	113
10.3.2.2.	Conclusion.	117
10.2.3.	Comparaison des règles de production avec le schéma Entité/Association.	118
Section 10.4 : Les réseaux sémantiques.		120
10.4.1.	Définition des réseaux sémantiques.	120
10.4.1.1.	Eléments du modèle. Définitions	120
10.4.1.2.	Inférences.	121
10.4.1.3.	Relations plus que binaires	123
10.4.1.4.	Remarques	125
10.4.1.5.	Raisonnement.	126
10.4.1.6.	Critique générale	127
10.4.2.	Application des réseaux sémantiques au modèle juridique	128
10.4.2.1.	Schéma conceptuel	128
10.4.2.2.	Conclusion.	132
10.4.3.	Comparaison des réseaux sémantiques avec le schéma Entité/Association.	133

Section 10.5 : Les frames et les scripts.	136
10.5.1. Définition des frames et des scripts.	136
10.5.1.1. Définitions	136
10.5.1.2. Inférences.	139
10.5.1.3. Critique générale	139
10.5.1.4. Les scripts	140
10.5.2. Application des frames au modèle juridique. . .	141
10.5.2.1. Schéma conceptuel	141
10.5.2.2. Raisonnement.	147
10.5.2.3. Mises à jour.	148
10.5.2.4. Conclusion.	149
10.5.3. Comparaison des frames avec le schéma Entité/Association.	150
Section 10.6 : Conclusion	152
CONCLUSION.	154

INTRODUCTION

Ce mémoire traite d'un domaine particulier d'application de l'informatique : le droit. Ceci peut paraître étonnant, dans la mesure où la décision judiciaire semble peu appropriée à un traitement automatisé.

Pourtant, l'informatique peut aider le juge ou l'avocat dans ses choix. Il ne s'agit pas de demander à l'ordinateur de prendre lui-même la décision, ce qui serait aberrant, mais bien de donner au juriste le plus d'éléments possible au sujet du cas étudié, lui permettant de rendre un jugement valide et appuyé.

Nous parlerons dès lors d'un "système informatique d'aide à la décision" (SIAD), judiciaire en l'occurrence. La définition d'un SIAD pourrait être la suivante (Ginzberg et Stohr) : "C'est un système d'information basé sur les technologies de l'information, destiné à apporter une aide aux tâches de processus décisionnels pour lesquels il n'est ni possible, ou ni souhaitable, d'automatiser l'entièreté du processus de décision."

Le C.R.I.D. (Centre de Recherche en Informatique et Droit) des Facultés Universitaires Notre-Dame de la Paix à Namur) effectue depuis quelques années des recherches pour tenter de réaliser un système de ce type. Dans ce cadre, deux mémoires de fin d'études ont été développés : le premier [HICK 82] réalise une structuration du problème et aboutit à un schéma conceptuel du jugement, le second [PONC 84] traite les côtés matériel et logiciel du problème et implémente un prototype.

Trois aspects ont été retenus comme facilitant la prise de décision judiciaire : l'aspect documentaire, signifiant la consultation de la doctrine et de la jurisprudence, l'aspect comparatif, considérant les jugements similaires rendus dans la matière visée, et l'aspect évaluatif, permettant d'évaluer la cohérence interne et externe d'un jugement.

Ce mémoire fait le point sur le résultat obtenu et tente de revoir le problème juridique dans un but d'amélioration et d'extension du prototype.

La recherche réalisée aux Facultés de Namur n'est pas un effort isolé. De plus en plus de juristes, d'informaticiens et de logiciens s'intéressent à ce problème. Les travaux s'étendent de la création de bases de données juridiques à l'étude de logiques

non classiques relatives à la notion d'obligation (logique déontique), en passant par l'analyse des aspects factuels et temporels dans la formalisation et l'automatisation de la décision judiciaire [PROC 85].

C'est donc le signe qu'il y a un besoin réel à combler, et qu'il convient d'y répondre le mieux possible.

Au stade où en était le prototype, il était nécessaire de revoir ses possibilités, de façon à rencontrer au mieux les désirs d'un juriste confronté à un litige. Par conséquent, l'amélioration à lui donner devait porter plus sur le fond que sur la forme.

Nous sommes donc partis de l'hypothèse qu'il fallait trouver de nouvelles fonctions d'aide à la décision, et pour ce faire, qu'il fallait restructurer les éléments intervenant dans le jugement.

Dans cette perspective, ce mémoire développe deux modèles de la décision et de l'argumentation juridiques, qui, après comparaison, nous permettront de retenir les quelques éléments qui serviront à proposer un prototype d'aide à l'argumentation juridique comme extension complémentaire aux fonctions documentaires et évaluatives déjà réalisées.

Le système obtenu donnera lieu à une étude des modes de représentation des connaissances à l'usage dans les systèmes experts, de manière à trouver un formalisme mieux adapté aux besoins de l'argumentation juridique.

Le travail sera divisé en trois parties :

La première partie resitue le projet conceptuellement et fonctionnellement (Chapitre 1), envisage les différentes possibilités d'extension (Chapitre 2) dans la perspective de développement du système, en fonction des limites observées, donne et justifie le choix d'élargir le modèle juridique et de rechercher un nouveau formalisme de représentation des connaissances (Chapitre 3).

La deuxième partie étudie donc le problème juridique en présentant le modèle théorique de la décision judiciaire proposé par le Professeur J. Wroblewski (Faculté de droit, Université de Lodz, Pologne) (Chapitre 4) et le modèle empirique de l'argumentation juridique développé par l'équipe du C.R.I.D. à Namur (Chapitre 5). Elle identifie les liens et les différences entre ces deux modèles (Chapitre 6) et propose une extension du schéma conceptuel (Chapitre 7) et des fonctions (Chapitre 8) d'aide à la décision judiciaire sur base du second modèle.

La troisième partie est un examen critique du système obtenu (Chapitre 9) et de la représentation conceptuelle des données (Chapitre 10). En fonction des limites retenues, elle définit et analyse les formalismes de représentation des connaissances connus dans les systèmes experts (Chapitre 11) et justifie la préférence donnée à un mode de représentation (Chapitre 12).

PREMIERE PARTIE :

ENVIRONNEMENT DU MEMOIRE :

PROTOTYPE D'AIDE A L'ELABORATION DE JUGEMENTS.

Le but de cette première partie est de resituer le projet de Système d'Aide à la Décision (SIAD) judiciaire développé par l'équipe "Logique et Droit" du Centre de Recherche en Informatique et Droit (CRID) des Facultés Universitaires Notre-Dame de la Paix à Namur.

Le premier chapitre est consacré aux objectifs du projet et à la solution conceptuelle et fonctionnelle du prototype obtenu. Le deuxième chapitre énonce et critique quelques extensions possibles à apporter au prototype actuel et le troisième chapitre choisit de développer une de ces extensions et justifie ce choix.

CHAPITRE 1 : Présentation d'une synthèse du prototype actuel. [BODA 85]

Section 1.1 : Objectifs du système.

Dans le but de situer le travail réalisé dans ce mémoire, il convient tout d'abord de rappeler les objectifs poursuivis lors du développement du prototype de système d'aide à la décision judiciaire implémenté aux Facultés de Namur.

Que peut apporter ce système à un juriste ? Son objectif premier est de constituer pour le juge ou l'avocat une aide à l'élaboration de jugements, en lui fournissant, d'une part, une documentation détaillée de la jurisprudence se rapportant à une matière juridique déterminée, d'autre part, une analyse des raisonnements suivis dans ces jugements.

Il s'agit donc d'offrir une recherche documentaire sur une base de données reprenant un ensemble, le plus complet possible, de jugements rendus dans des situations apparentées au cas traité.

L'analyse de cette jurisprudence doit permettre d'y découvrir d'éventuelles anomalies, contradictions ou incohérences. Ces tests seront de même appliqués sur des projets de décisions introduits de manière interactive dans le but d'évaluer leur validité en fonction de certains critères choisis.

La consultation d'un tel système tente donc de répondre aux problèmes particuliers posés par la prise de décision judiciaire, en aidant le juriste à construire et à évaluer son argumentation.

Section 1.2 : Solution conceptuelle.

1.2.1 Les éléments d'un jugement.

Pour établir la base de jugements qui alimente les traitements, il est nécessaire de repérer les éléments intervenant dans une décision. On en distingue deux types :

- les **informations structurées**, renseignements généraux relatifs au jugement ;
ex : ville, date, résultat.
- les **informations non structurées a priori**, qui constituent le raisonnement : éléments de fait et de droit considérés par le juge comme arguments de la décision.

Ces éléments de droit seront par la suite toujours appelés les "**standards**".

"Le standard est une fonction décrite par l'autorité, dont la présence est affirmée ou rejetée, fonction par laquelle le juge associe à un certain nombre de faits structurés ou non, un certain effet juridique prédéterminé." (Yves Poullet)

Les faits et les standards possèdent des propriétés semblables :

A l'état brut, le fait ou le standard est caractérisé par une **polarisation**, c'est-à-dire qu'il a, a priori, un effet juridique déterminé. On considérera, par exemple, qu'il joue toujours soit en faveur de l'inculpé, soit en sa défaveur, soit qu'il a un effet neutre pour n'importe quelle décision.

Lorsqu'un fait ou un standard intervient dans un jugement, il s'y trouve à l'état affirmé ou nié.

De même, le juge lui attribue une certaine influence sur la décision. Il considérera que cette influence est nulle, subsidiaire ou importante, selon les cas. Cette troisième propriété est appelée la **pondération**.

1.2.2 La formalisation du raisonnement du juge.

Une fois ces différents composants du jugement énumérés, il reste à trouver la façon dont ils sont agencés pour obtenir un résultat.

Le juge dispose de faits, y applique des standards, et désire aboutir à une décision. La forme que l'on donnera à son raisonnement est le plus naturellement celle du SI...ALORS... :

SI F1, F2, ..., Fn, ALORS DECISION

où les Fi sont des éléments de fait ou de droit.

Ce raisonnement se décompose en général en expressions du type :

SI F1, F2, ..., Fm, ALORS S1 ;

SI F1, F1+1, ..., Fp, ALORS S2 ;

etc.

où les Fi et Si sont des éléments de fait et de droit,
ou d'autres expressions.

Ces expressions représentent chacune un pas supplémentaire pour arriver à la décision finale. De telles implications sont appelées **compositions**.

Le plus souvent, ce sont des compositions de faits impliquant l'application du standard correspondant, mais l'on peut trouver des implications entre faits ou entre standards uniquement.

L'antécédent ou le conséquent peut lui-même être une autre composition, alors appelée respectivement **composition indirecte** ou **composition dépendante**. Mais dans la pratique, on s'aperçoit que le conséquent se réduit le plus souvent à un standard, et plus rarement à un fait.

Le formalisme utilisé implicitement dans cette analyse est celui de la **logique propositionnelle**. Celle-ci possède l'avantage de clarifier le langage naturel, de le rendre univoque après formalisation, et de permettre ensuite une étude de la validité d'un jugement basée sur les tables de vérité des opérateurs logiques.

Le concept de composition est toujours lié à un **effet juridique**, qui peut être comparé sémantiquement à la propriété de polarisation des faits et standards.

C'est ce cadre conceptuel et logique qui va guider le juriste dans son analyse de chaque jugement, pour en retirer une structure univoque qui servira de base aux traitements.

1.2.3 Problème juridique étudié.

Le projet de système d'aide à la décision judiciaire tel qu'il est développé aux Facultés de Namur se limite à un problème juridique déterminé, celui du **cautionnement**, réduit lui-même à un cas particulier. En quelques mots, en voici la description :

Une société, appelée par la suite **débiteur principal**, ouvre un crédit toutes sommes et à durée illimitée dans une banque, le **créancier**. Une personne jouant un rôle important dans cette société, soit comme administrateur-délégué, soit comme actionnaire principal, la **caution originaire**, se porte garant pour le remboursement de ce crédit devant la banque. C'est le **contrat de cautionnement**.

Cette personne quitte un jour ses fonctions importantes dans la société. Elle y est remplacée par la **caution nouvelle**, qui apporte, elle aussi, sa garantie à la banque pour le même crédit.

Quelques temps plus tard, la société se voit dans l'obligation et, malheureusement, l'impossibilité de rembourser la banque. Le problème qui se pose alors est de savoir si la caution originale doit toujours satisfaire aux obligations du premier contrat de cautionnement : Doit-elle encore rembourser la banque ?

Le rôle du juge dans ce conflit sera entre autres de déterminer s'il y a eu de la part de la caution originale une renonciation explicite ou implicite à ses obligations dans le contrat.

Les différents juges confrontés à un tel cas donnent aux requêtes du créancier des solutions parfois radicalement opposées, tout en disposant d'éléments de preuve similaires. C'est donc un domaine tout à fait approprié à une analyse en profondeur de la jurisprudence, de façon à y déceler les composantes du raisonnement qui font basculer la décision et/ou qui éventuellement la rendent invalide.

1.2.4 Le schéma conceptuel.

Le modèle Entité/Association [BODA 83] a été choisi pour représenter les concepts généraux du jugement et ceux particuliers au problème du cautionnement.

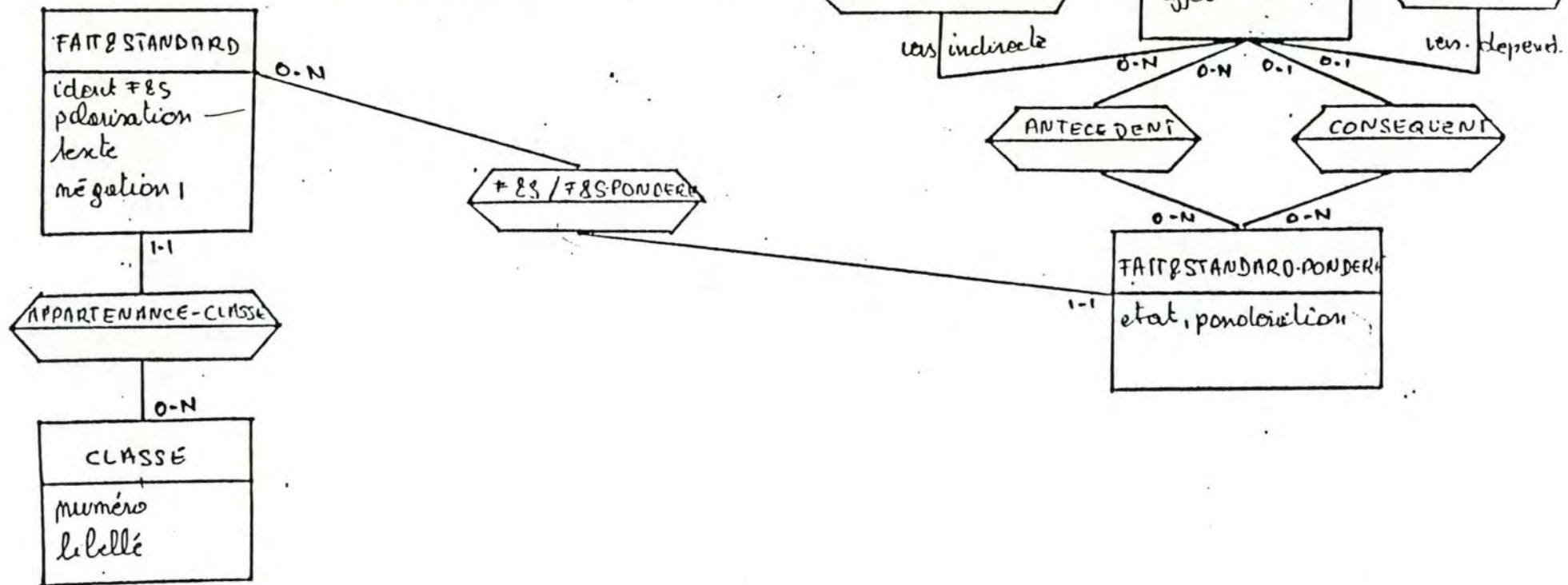
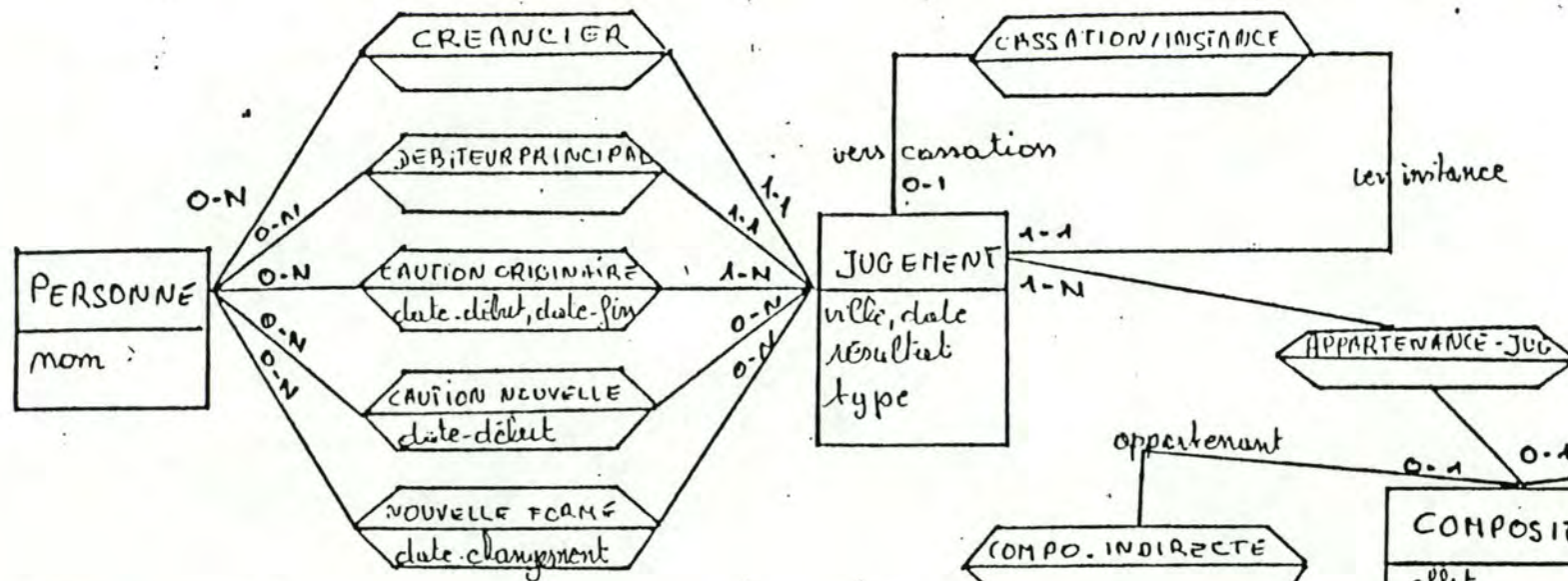
Dans ce cadre précis, les valeurs attribuées aux concepts de polarisation d'un fait ou standard et d'effet juridique d'une composition sont les suivantes :

Un fait ou standard aura une polarisation **positive** s'il joue en faveur de la libération des obligations de la caution originale, **négative** s'il joue en faveur du maintien de la caution, ou **neutre** s'il n'a aucun rôle a priori.

L'effet d'une composition sera **positif** si elle constitue un argument en faveur de cette libération, **négatif** dans le cas contraire.

SCHEMA CONCEPTUEL : [HICK 82]

;



DESCRIPTION DU SCHEMA CONCEPTUEL : [HICK 82]

Les types d'entités.

PERSONNE :

Une personne est une personne physique ou morale qui intervient dans un jugement.

"nom" est le nom de la personne physique ou la raison sociale de la personne morale.

JUGEMENT :

Un jugement est une décision juridique relative au problème de cautionnement restreint au cas décrit plus haut.

"ville" est la ville où a été rendu le jugement.

"date" est la date du jugement.

"type" représente le degré de juridiction. Ses valeurs sont Cassation et Instance.

"résultat" donne le résultat du jugement. Si celui-ci est de type Instance, les valeurs du résultat sont Maintien et Libération suivant que la caution originaire est maintenue ou libérée de ses obligations envers le créancier. Si le jugement est de type Cassation, les valeurs du résultat sont Cassé et Identique suivant que la décision d'instance correspondante est cassée ou maintenue.

"texte" est le texte de jurisprudence.

FAIT&STANDARD :

Un fait&standard est un élément qui peut intervenir dans la formalisation d'un jugement. Cet élément est soit un fait, soit un standard.

"nrofstd" est l'attribut identifiant du fait ou du standard.

"polarisation" donne la polarisation de fait ou du standard.

Ses valeurs sont Positive, Négative et Neutre.

"texte" est le libellé du texte du fait ou du standard.

"texte-négation" est le libellé du fait ou du standard à l'état nié.

CLASSE :

Une classe regroupe des faits&standards relatifs à un même concept. Tous les standards sont regroupés dans une même classe.

"numéro" est l'attribut identifiant de la classe.

"libellé" est une brève description du contenu de la classe.

COMPOSITION :

Une composition représente en tout ou en partie l'argumentation juridique développée par le juge dans un jugement.

"négation" dit si la composition est une négation conditionnelle ou une conditionnelle.

"effet" est l'effet de la composition. Ses valeurs sont + et - suivant que l'effet de la composition est de libérer ou maintenir les obligations de la caution originale.

FAIT&STANDARD PONDERE :

Un fait&standard pondéré est un fait ou un standard qualifié auquel le juge a donné dans un jugement, une pondération et un état et qui peut donc intervenir dans une composition.

"état" est l'état du fait ou du standard. Ses valeurs sont Affirmé et Nié suivant que le fait ou le standard est présent ou non.

"pondération" est la pondération du fait ou du standard. Ses valeurs sont Important, Subsidaire et Sans importance suivant le degré d'importance attribué par le juge pour le fait ou le standard dans la composition.

Les types d'associations.

CREANCIER (PERSONNE , JUGEMENT) :

une personne joue le rôle de créancier dans un jugement.

DEBITEUR-PRINCIPAL (PERSONNE , JUGEMENT) :

une personne joue le rôle de débiteur principal dans un jugement. La personne est une personne morale.

CAUTION-ORIGINAIRE (PERSONNE , JUGEMENT) :

une personne joue le rôle de caution-originaire dans un jugement.
La personne est une personne physique.

"date-début" est la date du début de l'engagement de la personne
comme caution originaire.

"date-fin" est la date de la fin de l'engagement de la personne
comme caution originaire.

CAUTION-NOUVELLE (PERSONNE , JUGEMENT) :

une personne joue le rôle de caution-nouvelle dans un jugement.
La personne est une personne physique.

"date-début" est la date du début de l'engagement de la personne
comme caution nouvelle.

NOUVELLE-FORME (PERSONNE , JUGEMENT) :

une personne joue le rôle de la nouvelle forme de la société
débiteur principal dans un jugement. La personne est une
personne morale.

"date-changement" est la date du changement de forme de la
société débiteur principal.

CASSATION/INSTANCE (JUGEMENT , JUGEMENT) :

relation qui associe un jugement de type Instance au jugement de
type Cassation qui lui correspond.

APPARTENANCE-JUG (JUGEMENT , COMPOSITION) :

une composition a été développée dans un jugement.

COMPO-INDIRECTE (COMPOSITION , COMPOSITION) :

une composition est composition indirecte c-à-d l'antécédent
d'une autre composition.

COMPO-DEPENDANTE (COMPOSITION , COMPOSITION) :

une composition est composition dépendante c-à-d le conséquent d'une autre composition.

ANTECEDENT (FAIT&STANDARD PONDERE , COMPOSITION) :

un fait ou standard pondéré est un des antécédents d'une composition.

CONSEQUENT (FAIT&STANDARD PONDERE , COMPOSITION) :

un fait ou standard pondéré est le conséquent d'une composition.

F&S/F&S-PONDERE (FAIT&STANDARD , FAIT&STANDARD PONDERE) :

un fait ou standard a reçu une pondération dans un jugement pour donner un fait ou standard pondéré.

APPARTENANCE-CLASSE (FAIT&STANDARD , CLASSE) :

un fait ou standard appartient à une classe.

Contraintes d'intégrité.

1. Un FAIT&STANDARD PONDERE doit intervenir dans au moins une COMPOSITION soit comme ANTECEDENT, soit comme CONSEQUENT.
2. Une COMPOSITION n'a qu'un seul conséquent. Celui-ci peut être un FAIT&STANDARD PONDERE ou une COMPOSITION qui est dite DEPENDANTE. Les types d'associations CONSEQUENT et COMPO-DEPENDANTE sont donc exclusifs.
3. Une COMPOSITION a au moins un antécédent. Celui-ci peut être un FAIT&STANDARD PONDERE ou une COMPOSITION qui est dite INDIRECTE. Pour chaque COMPOSITION, il faut donc au moins une association ANTECEDENT ou une association COMPO-INDIRECTE.
4. Une PERSONNE intervient au moins dans une des associations suivantes : CREANCIER, DEBITEUR-PRINCIPAL, CAUTION-ORIGINAIRE, CAUTION-NOUVELLE, NOUVELLE-FORME.

A partir de ce modèle conceptuel, une base de données reprenant des jugements relatifs au cautionnement a été créée selon les règles de transformation de schéma E/A en BD relationnelle.

L'exploitation de cette base de données est décrite à la section suivante.

Section 1.3 : La solution fonctionnelle et sa mise en oeuvre.
[PONC 84]

Pour atteindre les objectifs cités en 1.1, le prototype réalise deux types de fonctions : les **fonctions documentaires** et les **fonctions d'évaluation**.

1.3.1 Fonctions documentaires.

Les fonctions documentaires offrent à l'utilisateur des possibilités de consultation et de mise à jour de la jurisprudence.

La consultation de la base de jugements s'effectue d'une part, sous forme de listes prédéfinies et/ou sélectives ;

Exemples : Liste des identifications de jugements,
Liste des compositions d'un jugement donné,
Liste des faits relatifs à une personne donnée,
Liste des jugements reprenant un raisonnement donné
sous forme d'une composition.
Etc.

Le dernier exemple cité de liste prédéfinie constitue une particularité très importante du système qui nous occupe, car cette forme d'interrogation de la base de jugements se fonde non pas sur un critère simple, mais bien sur une représentation logique composée du raisonnement.

D'autre part, le prototype offre des fonctions de recherche documentaire par mots-clés. Les textes des faits et standards, ainsi que les résumés de jugements, sont indexés sur base de certains mots considérés comme significatifs, qui serviront de mots-clés. L'utilisateur interroge la base de jugements par des expressions booléennes reliant des mots. Le système lui renvoie la liste des textes indexés par ces descripteurs, en prenant en compte, sur demande de l'utilisateur, l'analogie ou la synonymie.

Exemple : Liste des jugements indexés par les mots "CAUTION" OU "CONTRAT" ET "RENONCIATION".

1.3.2 Fonctions d'évaluation.

Comme annoncé dans la section 1.1, le système doit offrir à l'utilisateur la possibilité de tester des jugements existants ou des projets de décisions, sur base de critères de validité prédéterminés.

Le prototype actuel propose deux tests de cohérence interne d'une décision et un test de cohérence externe.

1.3.2.1 Cohérence interne.

"La cohérence interne d'un raisonnement s'attache à vérifier que la formalisation de la règle de droit développée par le juge dans un jugement est logiquement cohérente."

Elle est étudiée en confrontant les éléments internes à la décision, sur base des propriétés de polarisation et de pondération décrites à la section 1.2.

1.3.2.1.1 Test de polarisation.

La décision soumise au test de polarisation se voit attribuer une présomption d'incohérence lorsqu'un fait ou standard polarisé **positivement** intervient dans une composition dont l'effet juridique est le **maintien** de la caution originaire.

De même, lorsqu'un fait ou standard polarisé **négativement** intervient dans cette décision dans une composition dont l'effet juridique est la **libération** de la caution originaire, le jugement est également présumé invalide.

Dans la pratique, ce test consiste en fait à comparer la polarisation de chaque antécédent et celle du conséquent pour voir si elles sont compatibles. Si elles sont opposées, le test déclare la composition, et donc le jugement invalide.

1.3.2.1.2 Test de pondération.

Le test de pondération, adjoint au test de polarisation, vérifie que la présomption d'incohérence s'explique par la pondération attribuée par le juge au fait ou standard qui en est la cause.

Un fait auquel le juge a accordé une importance nulle ne peut rendre une décision entièrement invalide. Dans ce cas, le test de pondération ne justifie plus la conclusion tirée du test de polarisation. Il la renforcerait dans le cas où le fait serait considéré comme important aux yeux du juge.

1.3.2.2 Cohérence externe.

La cohérence externe d'un jugement est testée par rapport à des éléments de référence extérieurs définis par les juristes. Des relations d'implication, d'équivalence et d'incompatibilité sont reconnues comme existant a priori entre deux faits ou standards. Les couples ainsi définis sont appelés **groupes de validité**.

Le test consiste alors à vérifier que dans un même jugement, les compositions ne mettent pas deux faits ou standards en relation incompatible avec un groupe de validité.

Par exemple, un juge qui utiliserait dans son argumentation à la fois le standard de la cause et celui de l'acte abstrait, rendrait par là-même sa décision incohérente.

1.3.3 Implémentation.

Le dialogue entre l'utilisateur et le prototype se déroule grâce aux techniques de menus en cascade, de questions-réponses et de formats prédéfinis.

Le système est implanté actuellement sur IBM_PC ou compatible.

L'ensemble des fonctions documentaires a été implémenté grâce au SGBD DBASEII, ainsi que les tests de validité interne.

Le test de validité externe, après avoir été développé en PROLOG sur DEC2060, est maintenant implémenté en MICRO-PROLOG sur le PC.

Section 1.4 : Limites du prototype actuel.

1.4.1 Limites de modèle sémantique.

- Les faits sont faiblement structurés. Ils sont uniquement regroupés sous la notion de classe, trop générale pour être efficacement utilisée. De plus, ils sont retenus indépendamment de leur contexte socio-économique, géographique et temporel.
[HICK 82]

- Les valeurs des propriétés de polarisation et de pondération sont attribuées aux faits et standards par un expert juriste, sans critère précis et objectif, et sont donc le fruit d'une appréciation subjective.

- Le système actuel ne s'applique qu'au problème particulier du cautionnement, et ne permet pas de présumer de la généralité de la démarche. [BODA 85]

1.4.2 Limites au contrôle de cohérence et de validité.

- Les hypothèses posées pour la définition de la "non-validité" ne font pas l'objet d'un accord unanime parmi la communauté des juristes, car, pour certains, il est impossible, et même faux, de définir des relations logiques a priori entre des éléments de fait ou de droit.

- Les tests actuels ne comparent pas les jugements entre eux.

1.4.3 Limites de l'implémentation.

- La réalisation du test de validité externe a été effectuée dans un langage différent de celui des autres fonctions du système. L'utilisateur ne dispose donc pas d'une interface commune pour toutes les exploitations possibles.

- L'implémentation ne répond pas à des critères de performance acceptables en ce qui concerne le temps de réponse, vu les restrictions imposées par le logiciel utilisé. [PONC 84]

CHAPITRE 2 : Extensions envisagées.

Nous examinons ici quelques possibilités d'extension du système existant, en réponse aux limites soulevées à la section 1.4.

Section 2.1 : Extension des requêtes.

Actuellement, les fonctions documentaires du prototype proposent des requêtes sous forme de listes prédéfinies et sélectives.

Il est toujours possible d'augmenter les possibilités d'interrogation, en "prédéfinissant" d'autres listes dont les critères de sélection sont les propriétés existantes.

Exemple : Liste des jugements rendus dans une ville donnée.

Reste à savoir si ces nouvelles requêtes apporteraient beaucoup dans une perspective d'aide à l'élaboration de jugements ! Sans doute a-t-on choisi jusqu'à présent celles qui paraissaient le mieux appropriées à cette exigence, et serait-il superflu d'en créer de nouvelles sur base de ces mêmes propriétés.

Une évolution importante serait d'offrir à l'utilisateur la possibilité d'interroger la base en se servant d'un langage proche du langage naturel, mais réduit à une syntaxe et un vocabulaire donnés.

Ce langage permettrait de sélectionner les données selon des combinaisons choisies au moment même par l'utilisateur, et non plus limitées à des requêtes prédéfinies.

Par contre, un élargissement de la structure sémantique des faits et standards, adapté aux besoins des juristes permettrait d'introduire de nouveaux critères d'interrogation de la base de jugements.

Jusqu'à présent, les notions de temps et d'espace, par exemple, sont restées absentes de la sémantique du modèle, alors qu'elles auraient pu être exploitées.

En particulier, la notion de temps pourrait orienter l'examen de la jurisprudence. En effet, l'interprétation donnée à la loi évolue selon le temps, le texte des standards évolue donc également.

De même, la polarisation des faits change avec l'époque, le lieu ou les circonstances : Un fait considéré comme répréhensible il y a plusieurs années, dans un pays donné ou en temps de guerre, peut l'être beaucoup moins à l'heure actuelle, dans un autre pays ou en temps de paix.

Il serait très certainement utile pour l'utilisateur de sélectionner les faits et standards relatifs à une époque, un lieu et des circonstances données, avant de poursuivre sa recherche par des requêtes basées sur les critères précédemment exploités.

Voici quelques exemples de questions à introduire :

"Liste des faits considérés avant (après) une date donnée, et leur polarisation."

"Evolution de la polarisation d'un fait selon le temps."

"Evolution de la pondération d'un fait selon le temps."

"Liste des standards utilisés dans un pays donné."

De telles interrogations deviendront réellement intéressantes pour le juriste lorsqu'on étendra la base de jugements au niveau international et à d'autres domaines que celui du cautionnement.

Dans une perspective d'extension immédiate du prototype, leur utilité paraît moins évidente.

La définition d'un langage d'interrogation adapté nécessiterait une étude approfondie de ses besoins syntaxiques et sémantiques, que nous n'avons pas choisi d'aborder dans le cadre de ce travail.

Section 2.2 : Extension des tests.

Les tests de validité portent jusqu'à présent sur les compositions d'un jugement, sans examiner leurs interdépendances:

Dans le test de polarisation-pondération, chaque argument est comparé à l'effet juridique de la composition dans laquelle il intervient.

Dans le test de validité externe, la relation logique entre deux éléments d'une même composition est comparée à des groupes de validité extérieurs au jugement.

D'autres tests de cohérence interne pourraient vérifier la cohésion d'un jugement, en tenant compte de l'ensemble de ses compositions et de leurs liens logiques. Ce type de test est appelé **test inter-compositions**. [BODA 85]

Il peut être intéressant également de comparer des jugements entre eux par un **test inter-jugements**.

2.2.1 Tests inter-compositions.

2.2.1.1 Test des états.

Ce test consiste à déceler, pour chaque fait ou standard intervenant dans un jugement, si ce fait ou standard n'apparaît pas dans les différentes compositions de ce jugement dans des états incompatibles.

En effet, le juge ne peut à la fois affirmer et nier un fait, ou appliquer et rejeter un standard.

En conséquence, un jugement sera déclaré invalide lorsqu'un fait&standard y est présent dans les deux états **affirmé** et **nié**.

2.2.1.2 Test des relations.

Contrairement aux précédents tests de cohérence interne, le test des relations tient compte des liens logiques de conjonction, disjonction, implication, équivalence ou incompatibilité, entre les éléments de fait et de droit formalisés par l'expert juriste.

Une relation logique entre deux mêmes faits ou standards ne peut être contredite par la suite dans une autre composition du jugement.

Par exemple, on ne peut pas trouver

$$F1 \Rightarrow F2$$

dans une composition, et

$$F1 \mid F2 \quad \text{ou} \quad F1 \ \& \ \sim F2$$

dans une autre composition du même jugement.

La démarche est semblable à celle du test de validité externe, mais plutôt que de comparer les relations internes au jugement avec des relations définies a priori de façon externe, elle les compare mutuellement à l'intérieur même du jugement.

2.2.2 Test inter-jugements.

Un test inter-jugements consiste à comparer deux jugements dans le but d'y déceler une incohérence.

Le jugement réputé invalide est alors celui dont le niveau d'instance est le plus bas, ou celui à l'intérieur duquel une autre incohérence serait mise à jour par un test supplémentaire.

Les critères de compatibilité inter-jugements pourraient être les suivants :

- Deux jugements ayant à décider des mêmes faits, doivent aboutir à une décision identique.

Remarque : Dans cette optique, une décision cassée serait toujours réputée invalide.

- Les liens logiques entre faits et standards doivent répondre aux critères de cohérence du second test inter-compositions entre compositions de jugements différents.

Ces propositions d'extension des tests de validité ne sont certainement pas exhaustives. De plus, une structuration plus détaillée des faits et standards telle qu'il en est question à la section 1.1, pourrait amener à déterminer de nouveaux critères de cohérence d'une décision, et donc de nouveaux tests.

Deux remarques doivent cependant être faites :

Premièrement, pour chaque test envisagé, il faut que l'utilisateur dispose de tous les éléments pour en interpréter les conclusions, de façon à déterminer lui-même le degré de confiance qu'il peut lui accorder. Il serait malhonnête d'imposer à tous les utilisateurs une façon de juger l'incohérence d'une décision. Les informer des choix qui sont faits est indispensable dans le cadre des tests plus que partout ailleurs.

Deuxièmement, le prototype ne doit en aucune façon prétendre analyser complètement la cohérence d'un jugement. Une telle optique serait d'ailleurs illusoire, mais également indésirable, l'appréciation du juge en tant qu'être humain unique restant un de ses atouts majeurs.

Si l'on peut juger de l'incohérence d'un jugement par rapport à certains critères objectifs, jamais on ne pourra en présumer la parfaite validité, celle-ci impliquant des critères d'évaluation tout à fait subjectifs.

Section 2.3 : Interface fonctions documentaires - Test de validité externe.

Actuellement, lorsque l'utilisateur désire effectuer une session de travail complète sur le prototype, c'est-à-dire utiliser les fonctions de consultation, ainsi que les tests de validité interne et externe, il doit quitter l'environnement de DBase2 (consultation, tests de validité interne) pour entrer dans l'environnement de Micro_Prolog (test de validité externe).

Le système n'offre pas encore d'interface commune aux fonctions documentaires et d'évaluation.

Il existe des recherches qui tentent de développer de telles interfaces entre un langage tel que Prolog, et une base de données relationnelle [BIM 85] [JARK 84] [VANA 85] [VANH 85] [VASS 85] [VENK]. Il serait bon d'examiner les techniques proposées dans ce domaine, de façon à ne pas perdre les avantages respectifs des deux langages, tout en leur permettant de communiquer.

Le système offrirait ainsi un langage homogène d'exploitation alliant les possibilités d'accès à une BD relationnelle, les capacités logiques de Prolog, et les caractéristiques d'une interface conviviale.

Section 2.4 : Extension du modèle de raisonnement juridique.

L'objectif de cette section est de revoir la notion d'aide à la décision juridique par ordinateur, en fonction d'une meilleure appréhension du problème.

Nous ne pouvons aboutir à un résultat convenable sans remettre continuellement en question les choix préalables. Ceci peut être réalisé par un échange de points de vue avec des chercheurs qui travaillent au développement de systèmes semblables, et/ou qui en étudient les théories sous-jacentes.

Comme nous l'avons souligné, un système d'aide à la décision judiciaire doit fournir au juriste une aide à l'élaboration de jugements, en analysant la jurisprudence et en validant les décisions sur base d'un certain nombre de critères.

Dans cette optique, une première approche serait de mieux étudier la prise de décision judiciaire, dans le but de déceler tous ses intervenants.

Ceux-ci une fois mis en évidence, nous pourrions alors prétendre juger de la validité d'un jugement, car nous connaîtrions les éléments de fait et de droit à considérer, ainsi que les règles précises régissant leur utilisation et leurs interrelations.

Le Professeur Jerzy Wroblewski, professeur à l'université de Lodz, en Pologne, est un théoricien du droit. Une de ses oeuvres majeures est à ce propos une théorie complète de la prise de décision judiciaire [WROB 78].

Celle-ci analyse la motivation, au sens de la justification, de la décision du juge, en la décomposant en décisions d'interprétation, de preuve, etc., chacune étant prise en fonction d'éléments de fait, de normes, de directives et d'évaluations.

Une seconde approche est celle des chercheurs de l'équipe "Logique et Droit" du C.R.I.D., chargée de la réalisation du prototype [SCHA 86].

Elle consiste à se demander, de façon tout à fait pragmatique, quelle serait une aide réelle pour l'avocat ou le juge devant argumenter sa décision. "Un praticien pourrait être intéressé par un système lui permettant d'identifier l'argumentation la plus efficace, celle qui présente la plus grande probabilité d'atteindre le but poursuivi, qui est d'obtenir du juge une décision satisfaisante."

Pour atteindre cet objectif, il faut donner un modèle à l'argumentation juridique, modèle sur lequel le système se basera pour répondre aux questions du juriste, telles :

"Quel standard puis-je utiliser si je dispose de tels faits et que je désire plaider la libération de la caution originale ?",

ou

"Quels sont les faits à prouver si je veux défendre son maintien?".

Ce modèle établira une relation a priori entre la norme et les faits qu'elle nécessite de prouver.

Une telle approche constitue donc une extension du prototype actuel vers un prototype d'aide à l'élaboration d'argumentations judiciaires.

Ces deux modèles feront l'objet d'une étude plus détaillée dans les chapitres suivants.

Section 2.5 : Etude des formalismes de représentation des concepts utilisés.

La révision du modèle de raisonnement juridique pose à nouveau le problème de sa formalisation. Les concepts sont, jusqu'à présent, représentés sous la forme du modèle Entité/Association [BODA 83].

Après avoir étendu ces concepts, ce formalisme est-il toujours valide et efficace ? Ne peut-il pas être soumis à la critique, et comparé à d'autres techniques de représentation de connaissances ?

Les recherches en Intelligence Artificielle ont créé des formalismes tels que les règles de production, les réseaux sémantiques, les frames, etc. Il serait intéressant d'examiner ces modèles dans la perspective de formalisation du schéma conceptuel de la décision judiciaire.

De même, on peut se demander si la logique des propositions convient toujours pour développer le processus de raisonnement juridique. Ne pourrait-elle pas être remplacée par une autre logique ?

De nouveaux types de logiques sont actuellement étudiés en relation avec le cadre particulier du droit. Nous pensons ici à la logique déontique, mais aussi à la logique modale ou temporelle [BAUE 85] [BULY 85] [McCA 85] [SANC 85].

Sans nous attarder à ces logiques non classiques, il serait toutefois bon de soumettre à une analyse sérieuse la logique des prédicats comme structure du raisonnement du juge.

Il faut cependant noter que si la nécessité de revoir l'efficacité des formalismes précédemment choisis est réelle, elle ne remet pas en question leur conformité. Le modèle E/A éventuellement élargi est suffisamment riche pour représenter n'importe quelles notions ou relation, mais n'est peut-être pas le mieux adapté aux nouvelles structures du modèle.

CHAPITRE 3 : Décisions d'extensions immédiates.

Parmi les cinq propositions d'extension du prototype énumérées au chapitre 2, deux d'entre elles ont été retenues : L'extension du modèle de raisonnement juridique et l'étude des formalismes de représentation qui s'en suit.

Les autres possibilités ne sont pas abandonnées définitivement pour la cause, mais feront l'objet de travaux ultérieurs.

Section 3.1 : Elargissement du modèle du processus de décision et d'argumentation juridique.

Nous avons choisi de revoir le modèle du raisonnement judiciaire pour les raisons déjà citées à la section 2.4 ;

Il nous a paru utile de reprendre le problème à la base, c'est-à-dire de remettre en question la validité, mais aussi l'utilité d'un SIAD (Système Informatique d'Aide à la Décision) judiciaire.

L'analyse du professeur Wroblewski répond à ces exigences. Elle a été retenue car elle nous paraissait un outil utile pour fixer les concepts intervenant dans une prise de décision juridique, en permettant ainsi de trouver de nouveaux critères de validité d'un jugement. Elle pouvait aussi mettre en évidence les difficultés de formaliser la motivation de la décision judiciaire, et donc de juger de sa validité, puisque c'est bien de cela dont il s'agit jusqu'à présent.

La démarche empirique ayant pour objet l'argumentation juridique sera également exploitée et constituera un modèle servant de base à une extension du prototype d'aide à l'élaboration de jugements.

Ce premier choix s'explique aussi en regard des autres possibilités d'extensions qui étaient proposées, en ce sens que, si l'on parvient à revoir le modèle de la décision judiciaire, et, par le fait même, à mieux structurer les concepts existants, de nouvelles propriétés constitueront de nouveaux critères de sélection pour les requêtes et de validité pour les tests.

Ces extensions d'abord envisagées sont donc plutôt un prolongement logique de celle qui vient d'être choisie qu'une nécessité immédiate.

La création d'une interface commune aux fonctions documentaires et d'évaluation fera partie, de préférence, d'une phase d'amélioration technique du prototype, tenant compte de ses performances et de son caractère convivial, dans la perspective d'une utilisation effective par des juristes confrontés à des conditions réelles d'exploitation.

Les prochains chapitres tenteront de décrire, d'une part, le modèle théorique de Wroblewski, d'autre part, le modèle empirique de l'argumentation, d'analyser les conséquences de ces nouvelles

modélisations sur le prototype actuel et de choisir parmi ces développements possibles, celui qui paraît le mieux adapté aux objectifs d'un SIAD judiciaire.

Section 3.2 : Analyse des formalismes de représentation des connaissances.

Une fois les concepts choisis et les nouvelles fonctions développées, nous pourrions nous poser la question de la technique de représentation du modèle obtenu.

Il est nécessaire de trouver un formalisme mieux adapté à la nature à la fois des concepts et des fonctions, et de traiter sous ce formalisme commun des requêtes documentaires et des évaluations nécessitant des mécanismes d'inférence.

L'extension qui sera apportée au prototype nous permettra de mieux juger de l'utilité d'une telle analyse.

DEUXIEME PARTIE :

MODELES DES PROCESSUS

DE DECISION JUDICIAIRE ET

D'ARGUMENTATION JURIDIQUE.

Les objectifs de cette partie sont, d'abord, de décrire le modèle théorique de la décision judiciaire proposé par le Professeur J. Wroblewski (chapitre 4) et le modèle empirique de l'argumentation juridique de l'équipe du C.R.I.D. (chapitre 5), de les comparer ensuite pour en retirer d'éventuels points communs (chapitre 7), et enfin, de choisir un modèle et les caractéristiques de sa réalisation (chapitre 8).

CHAPITRE 4 : Modèle théorique de la décision juridique proposé par le professeur J. Wroblewski.

Section 4.1 : Motivation de la décision judiciaire. [WRO 78] [WROB 79]

4.1.1 La décision judiciaire.

La théorie générale des décisions permet d'analyser, en particulier, la décision judiciaire.

Selon cette théorie, les éléments constituant toute décision sont:

- le sujet,
- les alternatives du choix,
- le choix parmi ces alternatives selon des critères de préférence,
- les facteurs de situation, d'estimation, de raisonnement,
- la façon dont est exprimée la décision,

- la justification (arguments) de la décision.

Le sujet de la décision judiciaire est le juge, ou un corps collectif de juges.

Les alternatives proviennent de l'ambiguïté de la norme, qui doit alors être interprétée.

Les facteurs principaux de la décision judiciaire sont

- les informations du juge au sujet de la loi, des faits du cas cité ou d'autres cas,
- l'attitude épistémologique du juge, c'est-à-dire ses capacités cognitives, intellectuelles pour comprendre, sélectionner et combiner les informations dans le but d'en tirer les inférences,
- l'attitude évaluative du juge, liée à son attitude épistémologique, et à son idéologie de l'application du droit.

La décision judiciaire est exprimée de façon verbale.

4.1.2 La justification de la décision judiciaire.

4.1.2.1 Quand intervient la justification ?

La justification des décisions judiciaires apparaît au moins dans deux situations :

- lorsque la décision est prise par un corps collectif ayant des opinions divergentes,
- lorsqu'elle est un devoir imposé par la loi : la motivation légale.

Dans tous les autres cas, la décision doit toujours être "justifiable".

4.1.2.2 Types de justifications.

Une distinction peut être faite entre justifications

- **au sens le plus strict**, c'est-à-dire selon les règles d'une logique alétiqne (= logique classique des propositions ou des prédicats),
- **au sens strict**, c'est-à-dire selon une logique formelle plus riche (par exemple, la logique déontique, qui privilégie les notions d'obligation et d'interdiction),
- **au sens large**, c'est-à-dire non formalisable.

La décision judiciaire devrait être justifiée **au sens strict**, si l'on présuppose une logique formelle permettant l'introduction de normes et d'estimations.

De même, on parlera de justification

- **interne**, rapport entre prémisses (propositions, normes, estimations) et décision finale,
- **externe**, justification de la validité des hypothèses et règles de raisonnement utilisées.

4.1.2.3 La prise de décision judiciaire.

Pour établir un modèle de la justification de la décision judiciaire, il faut tenir compte des étapes nécessaires à cette prise de décision :

- Choix d'une norme valide ayant une signification suffisamment précise, et donc nécessitant souvent une interprétation.
- Décision de preuve dans le but d'établir l'existence des faits.
- Particularisation du fait sous la norme légale appliquée.
- Détermination des conséquences légales du fait selon la norme appliquée.

On aboutit à la formule suivante :

Selon la norme N,

valide dans le système légal LS, selon les critères de validité VC1, VC2, ..., VCn, étant donnés les faits F1, F2, ..., Fn, et les évaluations NE1, NE2, ..., NEn,

qui a le sens S, conformément aux directives d'interprétation DI1, DI2, ..., DI n, et aux estimations VI1, VI2, ..., VIn, nécessaires pour le choix et l'emploi de ces directives,

le fait F, dont l'occurrence à la place p et au temps t, a été prouvée par les moyens admis P1, P2, ..., Pn, conformément aux directives DP1, DP2, ..., DPn, et aux estimations VP1, VP2, ..., VPn,

a pour conséquences légales C1, C2, ..., Cn, conformément aux directives de choix de conséquences DCC1, DCC2, ..., DCCn, et aux estimations VC1, VC2, ..., VCn.

Par conséquent, quatre décisions doivent être justifiées :

- Décision de **validité**
- Décision d'**interprétation**
- Décision de **preuve**
- Décision du **choix de conséquences**.

4.1.2.4 Autres significations.

En plus de la justification et de la motivation légale de la décision judiciaire, deux autres interprétations de la motivation pourraient être analysées :

- L'**explication psychologique**, description des processus psychiques amenant le juge à la décision, et
- L'**explication sociologique**, considérant la décision judiciaire comme un fait social, élément d'un système complexe de relations sociales.

Ces deux dernières ne sont pas incluses dans la pratique de justification de la décision judiciaire.

Section 4.2 : Décision de validité.

4.2.1 Concepts de validité. [WROB 82]

4.2.1.1 Validité systémique.

La norme N est valide **systémiquement**, c'est-à-dire relativement à un système de référence (voir section 4.2.2 pour les différents types de systèmes légaux), si

- a) N est promulguée dans ce système,
- b) N n'est pas dérogée par des normes valides dans ce système,
- c) N est cohérente avec les normes valides dans ce système,
- d) si N n'est pas cohérente avec les normes valides dans ce système,
alors, soit elle ne perd pas sa validité d'après les règles acceptées de conflits de normes,
soit N est interprétée de telle façon qu'elle cesse d'être incohérente avec les normes en question.

4.2.1.2 Validité factuelle.

- Version radicale :

Une norme est valide si elle est utilisée, c'est-à-dire référée ou appliquée dans une décision disposant d'un cas.

- Version modérée :

Une norme est valide si elle remplit tous les critères de validité systémique

+ e) une norme remplissant les critères a), b), c), d) est valide seulement si elle est appliquée dans une décision opérationnelle.

4.2.1.3 Validité axiologique.

La norme légale incohérente avec des normes extra-systémiques (évaluations) déterminées, n'est pas valide.

4.2.2 Types de systèmes légaux et leurs concepts correspondants de validité. [WROB 85]

On détermine cinq types de systèmes légaux :

1. LSLE : ensemble des règles légales promulguées (validité systémique).
2. LSFC : normes valides systématiquement et leurs conséquences formelles.
3. LSIC : normes valides en LSLE et LSFC et leurs conséquences interprétatives.
4. LSOL : normes générales de LSLE, LSFC et LSIC et normes individuelles, c'est-à-dire les décisions (validité factuelle).
5. LSPP : principes, politiques et autres standards référés dans des normes générales de LSLE ou utilisés dans les décisions.

4.2.3 Formule.

N est valide dans un système légal LS

selon les critères VC1, VC2, ..., VCn

étant donné l'occurrence des faits F1, F2, ..., Fn
et des évaluations NE1, NE2, ..., NEn.

Section 4.3 : Décision d'interprétation. [WROB 69] [WROB 72]

4.3.1 Signification du mot interprétation.

- L'interprétation **au sens le plus large** est la compréhension d'objets culturels, l'entendement des sciences humaines.

- L'interprétation **au sens large** assigne une signification à certains complexes ou objets physiques, symboles ou signes.

- L'interprétation **au sens strict** permet d'éliminer un doute sur la signification propre d'un texte.

C'est cette dernière qui sera prise en compte lorsqu'on parlera d'interprétation juridique.

4.3.2 Directives d'interprétation (DI).

Dans les systèmes juridiques statutaires, la norme légale doit être utilisée selon sa "vraie" signification, qui doit être unique, avec un degré de précision suffisant pour les besoins de la décision légale.

Les directives d'interprétation doivent donc être formulées en fonction de cette nécessité.

Ces directives peuvent être divisées en deux groupes :

Directives du Premier Degré :

Elles déterminent la signification de la norme légale, par l'usage d'éléments de contexte fondamentaux :

- le langage : **DI linguistiques**
- le système légal : **DI systémiques**
- le contexte fonctionnel (économique, etc.) : **DI fonctionnelles.**

Exemples de directives du 1er degré se trouvant dans la loi :

"Ne jamais considérer le texte comme absent".

"Suivre la linguistique".

"Interpréter la norme de façon à ce qu'elle ne soit pas contradictoire avec d'autres normes du système".

"Suivre le but du législateur".

"Suivre le but de l'Etat".

"Suivre la notion de justice acceptée".

Directives du Second Degré :

Elles déterminent l'usage des règles du 1er degré :

- la séquence et les conditions d'application :

DI procédurales.

- la préférence en cas de conflit :

DI préférentielles.

Exemple :

"S'il y a un conflit entre les interprétations téléologique et linguistique, il faut préférer la première".

4.3.3 Modèle théorique de l'interprétation légale.

Le processus d'interprétation suit alors le schéma suivant :

1. Un doute apparaît sur la signification de la norme à appliquer au cas, parce qu'on constate :

A. Un manque de degré de précision suffisant des termes

ou B. Une contradiction au premier niveau avec d'autres normes

ou C. Une contradiction par rapport à certains buts, à une certaine notion de justice ou d'équité.

2. Utiliser les directives d'interprétation du 1er degré, selon celles du 2d degré.

3. Si les résultats sont différents selon différentes directives du 1er degré, effectuer le 4, sinon le 5.

4. Choisir entre les différents résultats, selon les règles du 2d degré préférentielles.

5. Formuler une décision interprétative établissant la vraie signification de la norme.

4.3.4 Formule.

La norme **N** a la signification **M**, dans le langage légal **J**, selon les directives d'interprétation **D1, D2, ..., Dn**, utilisées de manière déterminée par les valeurs acceptées **V1(D1), ..., Vn(Dn), V1(D2), ..., Vn(D2), ..., V1(Dn), ..., Vn(Dn)**.

4.3.5 Evaluations.

Chaque étape du modèle théorique nécessite des évaluations de la part du juge :

1. Doute.

Le juge doit évaluer le degré de précision des termes de la norme, ainsi que son éventuelle contradiction par rapport à certains buts ou notions de valeur.

2. Directives d'Interprétation.

Les conditions d'application de certaines DI impliquent la nécessité d'évaluer les critères qu'elles contiennent.

3. Types de valeurs.

Le premier choix qui revient au juge pour l'utilisation des DI est la poursuite de **valeurs statiques** ou **dynamiques**.

Les premières sont celles de la sécurité, de la certitude et de la stabilité légales.

Les secondes recherchent le maximum d'adéquation aux exigences de la vie socio-économique, idéologique et culturelle.

Selon cette distinction, les DI peuvent être groupées en **théories statiques**, donnant une signification constante de la norme, ou **dynamiques**, provoquant un changement de signification de la norme selon le contexte dans lequel elle est interprétée.

Il existe un groupe de directives d'interprétation communes aux théories statiques et dynamiques, mais qui sont insuffisantes pour former une théorie complète. Elles se fondent, par exemple, sur la "rationalité du législateur", le compromis entre légalité et justice.

Le choix de valeurs est donc une nécessité, mais limite la possibilité de rationalisation du processus d'interprétation.

4.3.6 Formalisation du processus d'interprétation. [WROB 69]

Il existe deux conceptions au sujet de la formalisation du processus d'interprétation :

La première prétend qu'il existe des systèmes de logique formelle utilisés comme références pour les raisonnements interprétatifs.

La seconde, au contraire, affirme que la logique en question n'est pas un calcul formel, mais un type de logique argumentative.

La première approche suppose qu'il est possible d'"interpréter" (utilisé dans le sens où l'on donne une interprétation à un modèle en le faisant correspondre avec les éléments du monde réel), certains calculs formels avec des expressions utilisées dans les raisonnements interprétatifs. Ceux-ci peuvent donc être formalisés grâce à des opérations logiques relativement simples :

A partir d'un ensemble de DI adéquat pour la détermination d'une signification douteuse, on les organise de manière cohérente, grâce à l'usage du "SI...ALORS...".

La seconde met en évidence les choix évaluatifs déterminant les hypothèses du raisonnement. L'utilisation d'une logique argumentative suppose donc la création d'une échelle de valeurs, opération quasi impossible, vu le nombre et la diversité de ces valeurs, selon les différents cas.

Dans les deux positions, la validité du raisonnement interprétatif ne peut être traitée que partiellement.

4.3.7 Modèle empirique de l'interprétation.

Ce modèle est une généralisation des descriptions

- **psychologique**, par l'analyse des mécanismes psychiques menant du doute à la décision, et
- **socio-psychologique**, par l'analyse du contexte social et psycho-social.

4.3.8 Le rôle des principes du droit dans la théorie et l'idéologie de l'interprétation juridique. [WROB 84]

4.3.8.1 Les types de principes du droit.

- Principe positif du droit (PJD) :

C'est une norme qui est explicitement formulée dans le texte du droit, c'est-à-dire une disposition légale, ou construite à partir des éléments contenus dans des dispositions légales, et qui est traitée comme plus importante que les autres, car jouant un rôle décisif dans l'institution juridique, dans l'acte normatif ou dans un système de droit.

- Principe implicite du droit (PJI) :

C'est une règle qui est traitée comme prémisse ou conséquence des dispositions légales ou normes, selon les règles de raisonnement admises dans la langue juridique.

- Principe extra-systémique du droit (PJE) :

C'est une règle, qui n'est ni PJD, ni PJI, non contenue dans le système du droit, provenant du droit comparé, ou des règles sociales acceptées dans la pratique judiciaire (morale, mœurs, droit naturel, politique, etc.).

- Principe-nom du droit (PJN) :

C'est un nom caractérisant les traits essentiels des institutions juridiques.

Exemples : Liberté contractuelle, vérité objective, bonne foi, etc.

Autrement dit, c'est un instrument de la description du droit en vigueur, mais ayant une importance normative.

- Principe-construction du droit (PJC) :

C'est une construction du législateur rationnel ou parfait présumée dans l'interprétation juridique.

Exemples : Cohérence, compétence linguistique, volonté du législateur.

4.3.8.2 Le rôle des principes dans la théorie de l'interprétation.

- Rôle des PJD :

Les PJD jouent un rôle, surtout dans le cas du conflit des lois et dans l'interprétation systémique, mais aussi dans l'interprétation linguistique :

- lorsque deux normes, dont un PJD, sont contradictoires, le PJD prévaut.
- le sens d'une expression linguistique, donné dans un PJD vaut dans les normes ordinaires.

- Rôle des PJI :

Le rôle des PJI est analogue à celui des PJD.

- Rôle des PJE :

Les PJE sont des moyens d'adapter la lettre de la loi aux besoins de la vie.

- Rôle des PJN :

Les PJN donnent une direction pour interpréter les normes de l'institution dont il met en relief les traits essentiels.

- Rôle des PJC :

Les PJC servent au fondement des directives d'interprétation et sont utilisés comme source des arguments pour l'interprète.

4.3.8.3 Le rôle des principes dans l'idéologie de l'interprétation.

- Dans l'idéologie statique :

Les PJD sont les plus utiles, car les plus stables.

Les PJI sont aussi importants car ce sont également des normes appartenant au système; mais la condition nécessaire à leur validité est le caractère strictement logique de leur construction.

Les PJE sont dangereux pour les valeurs statiques.

Les PJN et PJC sont utiles seulement s'ils sauvegardent les valeurs statiques.

- Dans l'idéologie dynamique :

Les PJD et PJI ont un rôle minime à jouer.

Par contre, les PJE sont très importants car ce sont des facteurs décisifs de la réalisation des valeurs dynamiques.

Les PJN et PJC jouent le rôle déterminé par leur contenu.

Section 4.4 : Décision de preuve. [WROB 81]

4.4.1 Que prouver ?

La preuve porte sur les faits qui interviennent dans le contexte du cas jugé.

Il faut prouver, pour chaque fait F, que

"Le fait F s'est produit au temps t et à l'endroit p".

4.4.2 Typologie des faits. [WROB 73]

4.4.2.1 Faits descriptifs ou évaluatifs.

Les faits **descriptifs** sont désignés comme des propositions existentielles.

Exemple : "Tuer un homme".

Les faits **évaluatifs** contiennent une évaluation.

Exemple : "Raison juste".

Ces derniers peuvent être décomposés en

1. L'établissement de l'existence de l'objet d'évaluation :

• "F existe en t et p".

2. L'évaluation :

"F a la valeur V".

4.4.2.2 Faits positifs ou négatifs.

"Le fait x déterminé positivement par le terme F dans la norme N" peut être traité comme une conjonction de la proposition existentielle :

" x existe en t et p"

et la proposition sémantique :

" x est F".

Les faits **négatifs** sont des expressions contenant "non" ou autres synonymes.

Si G est la classe des faits positifs,

F_n le terme négatif désignant G dans la norme N,

x pour le cas concret,

alors pour établir " x comme F_n existe en t et p" :

- a) " x existe en t et p"
- b) " x est G"
- c) "N désigne par F_n tout ce qui n'est pas F"
- d) "G est désigné par F_n dans la norme N".

4.4.2.3 Faits simples ou relationnels.

Un fait est **simple** si sa relation avec une norme n'est pas relevante pour établir son existence.

Dans le cas contraire, il est **relationnel**.

Exemple : "Contrat invalide".

"F existe en t et p" est analysé comme :

- a) "x existe en t et p"
- b) "x est dans la relation R à la norme N"
- c) "x est appelé «F» dans la norme N".

4.4.3 Les directives de preuve (DP).

Les **directives de preuve** sont soit des normes de loi, soit des règles extra-légales de la science et de l'expérience, telles la "conviction intime du juge" ou les expertises.

On parle de **règles de preuve légales** ou de **règles de preuve empiriques**, formant une **théorie légale** ou **formelle** de la preuve, ou une **théorie de la libre évaluation**.

Les **règles légales** déterminent les faits

- pouvant être objets de preuve
 - devant être démontrés
- ou - n'exigeant pas de preuve.

Elles réglementent l'admissibilité des moyens de preuve, et les présomptions légales.

Ces dernières peuvent être classées en trois catégories, selon qu'elles ont une source scientifique plus ou moins proche :

- les **présomptions para-empiriques**,

Exemple : Présomption de paternité sur le mari de la mère à la naissance, sauf preuve contraire.

- les **présomptions non-empiriques**,

Exemple : Présomption de la date de la mort du disparu d'un naufrage, fixée à l'heure prévue d'arrivée du bateau au port de destination.

- les présomptions contra-empiriques.

Exemple : Au Moyen Age, présomption de sorcellerie sur la femme qui ne coulait pas lorsqu'on la jetait dans l'eau.

4.4.4 Evaluations.

L'utilisation des directives de preuve est de nouveau liée à celle de valeurs.

Tout d'abord, comme vu plus haut, la preuve de faits évaluatifs et/ou relationnels comporte des jugements de valeurs.

La théorie de la libre évaluation oblige le juge à faire des choix évaluatifs : par exemple, il doit déterminer la validité d'une technique nouvelle ayant donné de nouveaux résultats d'expertise, différents des précédents, ou il doit choisir parmi les opinions d'experts divergentes, etc.

Les normes réglant l'inadmissibilité de certains moyens de preuve, tels les témoignages de personnes déficientes au niveau de leur perception, ou obtenus sous la menace, ou par d'autres moyens limitant la protection de la vie privée ou le secret professionnel, par exemple, impliquent des jugements à propos des valeurs de la personne, de la profession, des institutions ou de la sécurité juridique.

Le juge se trouve donc, à tout moment lors de la décision de preuve, confronté à des choix évaluatifs.

Aucune logique purement formelle ne peut contrôler ces décisions.

4.4.5 Formule.

Le fait F s'est déroulé au temps t et au lieu p,
selon les preuves acceptées P_1, P_2, \dots, P_n ,
basées sur les règles de preuve empiriques $RPE_1, RPE_2, \dots, RPE_n$,
et/ou sur les règles de preuve légales $RPL_1, RPL_2, \dots, RPL_n$,
et/ou sur les évaluations V_1, V_2, \dots, V_n .

4.4.6 Contrôle de la conviction intime du juge. [WROB 81]

4.4.6.1 Par la loi.

La loi peut contrôler les fautes telles que :

- les erreurs dans la détermination des faits,
- le manque d'explication de toutes les circonstances du fait,
- la contradiction entre les opinions du tribunal et le contenu des preuves.

4.4.6.2 Par la jurisprudence.

Les critères de contrôle sont les arguments de :

- **conséquence** : le tribunal doit être conséquent avec les preuves qu'il a acceptées ou rejetées.
- **totalité** : il doit considérer toutes les circonstances de l'affaire.

- la force convaincante : l'opinion de l'expert doit être complète, décisive, sinon elle doit être disqualifiée.

Ces contrôles sont donc basés sur la rationalité interne des décisions de preuve.

Section 4.5 : L'informatique comme aide au processus judiciaire. [WRO 72]

[WROB 70]

Une fois les composants de la décision judiciaire analysés, nous allons tenter, avec le Professeur Wroblewski, de les situer dans le processus judiciaire complet, en vue d'y appliquer l'informatique.

L'analyse du Professeur Wroblewski se base sur une prémisse de théorie du droit, à savoir qu'un procès n'est que l'application de la loi;

Dans un procès, le juge ne crée pas de normes juridiques, mais essaie de prononcer une sentence en s'appuyant sur des normes préexistantes. Les technologies de l'information peuvent avoir une utilité dans ce dernier acte du procès.

Il faut traiter le processus judiciaire comme un processus de transformation d'information. Si l'on peut construire son modèle informatif, alors on peut se demander si les ordinateurs, en tant que machines de traitement de l'information, ne pourraient pas être utilisés comme une aide pour retrouver, transformer ou élaborer des données relatives à la décision judiciaire.

4.5.1 Modèle du système d'information du processus judiciaire.

Soient C, le tribunal,

R, les informations sur la réalité, les faits du cas traité,

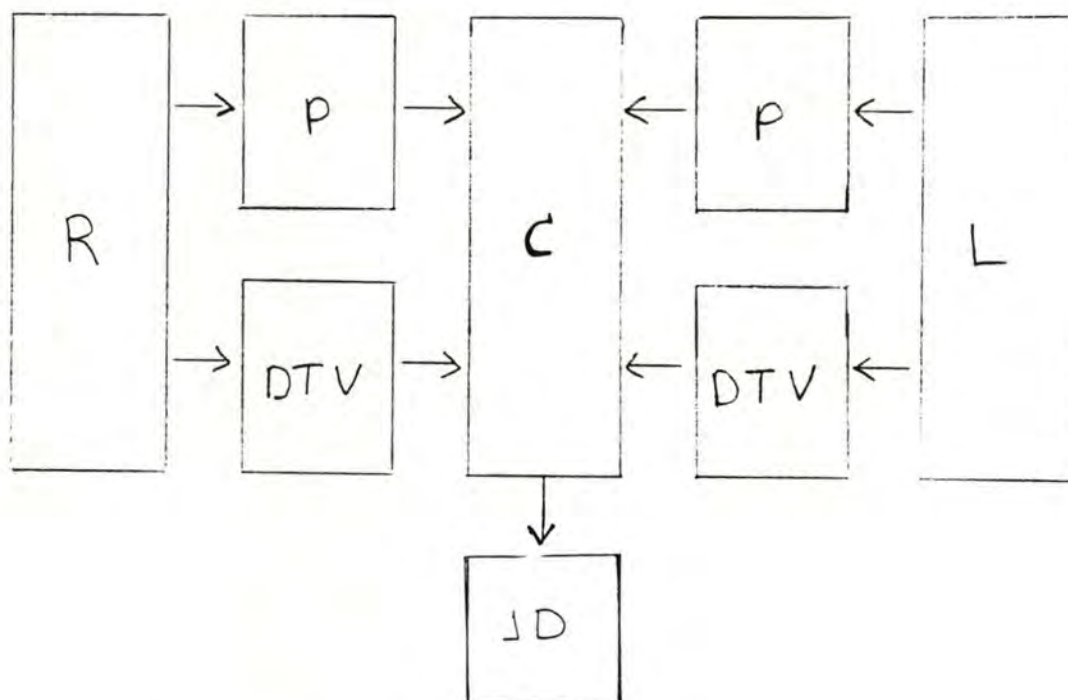
L, les informations au sujet des normes légales,

JD, la décision judiciaire,

P, les parties,

DTV, la dogmatique légale, la théorie légale, l'attitude évaluative propre au juge, conditionnée par l'idéologie acceptée des fonctions judiciaires,

on peut représenter le processus judiciaire selon le schéma suivant :



Les informations en entrée, R et L, n'arrivent pas brutes au tribunal. Elles sont soit présentées par les parties, qui les sélectionnent de façon à servir leur cause, soit prises par le tribunal, lui-même influencé par les DTV.

La transformation des informations en entrée en décision judiciaire est effectuée par les quatre opérations décrites dans les sections précédentes : les décisions de validité, d'interprétation, de preuve et de choix de conséquences.

4.5.2 Utilisation de l'informatique dans la prise de décision judiciaire.

Le modèle d'information ainsi défini, nous sommes maintenant capables de repérer les éléments pouvant faire l'objet d'un traitement informatique.

Les canaux de communication entre C et R et entre C et L, passant par P, peuvent être éliminés de ce modèle, car ils ne peuvent être informatisés.

Il reste à analyser les canaux de communication passant par DTV, pour chacune des opérations citées.

4.5.2.1 Décision de validité.

L'usage de l'informatique pour tester la validité d'une norme dépend des critères choisis :

- si les critères sont basés uniquement sur les propriétés syntaxiques des normes, sans référence à leur sémantique, il est possible de les automatiser, pourvu que les données légales soient présentées sous une forme rendant manifestes les contradictions.

Exemple : Présence de négations.

- si les critères impliquent des évaluations, la décision de validité est plus difficile à prendre mécaniquement.

Exemple : lex generalis - lex specialis.

4. 5. 2. 2 Décision d'interprétation.

L'usage de l'informatique pour établir la signification d'une norme est rendu complexe par l'élément évaluatif inhérent très important.

Cependant, plusieurs significations possibles pour chaque norme pourraient être introduites dans la mémoire. Mais la liste ne peut être complète, et les critères de choix entre les différentes possibilités requièrent des évaluations.

Tant qu'une échelle de valeurs n'est pas construite, l'ordinateur est limité par l'ensemble des significations inscrites dans la mémoire.

Cependant, la création d'une telle échelle apparaît illusoire dans le contexte institutionnel et culturel actuel. En effet, peut-on chiffrer des critères faisant intervenir des éléments psychologiques, socio-culturels ou politiques ? De même, rien ne laisse préjuger de la linéarité d'une telle échelle.

Quand bien même il serait possible de la définir, cette échelle rendrait le processus de décision trop rigide, et sans doute rapporté à une seule conception des valeurs, qui serait celle du formalisateur. Or, on ne peut imposer au juge un seul schéma d'évaluation.

4. 5. 2. 3 Décision de preuve.

L'informatique pourrait être utilisée lorsque les caractéristiques de l'élément de preuve le requièrent.

Exemple : Techniques d'identification d'objets.

4.5.2.4 Décision de choix de conséquences.

Plus que les trois autres, cette décision est une question de choix. Elle requerrait sur ordinateur un ensemble programmé de directives de choix, et de nouveau, la construction d'une échelle axiologique.

L'informatique n'est donc pas applicable à toutes les normes légales donnant des marges d'appréciation au juge.

4.5.3 Conclusions.

La difficulté principale rencontrée par l'informaticien est la construction d'échelles axiologiques adéquates, car elle entre en conflit avec plusieurs convictions concernant le caractère des valeurs légales et avec les idéologies des fonctions judiciaires dans nos cultures légales.

Mais le Professeur Wroblewski conclut son analyse en disant :

"Here and now we have enough to do to aid the judge with the most modern tools, that is with computers, in his decision-making, without absolving him from the responsibility for his choices."

Section 4.6 : Avantages et limites du modèle théorique.

Le modèle proposé par le professeur Wroblewski est une analyse du processus de décision en général, et de celui de la décision judiciaire en particulier.

Il décrit le squelette de base du jugement, autour duquel viennent se greffer les choix de valeurs du juge.

Il repère les informations et leurs règles de transformation avec précision et rigueur.

Cette analyse permet donc de donner une structure au jugement et de classifier ses composants selon de nouveaux critères.

La distinction qui est faite, pour chaque étape de la décision, entre les éléments formalisables et les évaluations, constitue une aide efficace dans une perspective de formalisation du jugement. Elle restreint la part de subjectivité du juriste aux évaluations et donne donc un squelette strict au jugement, qui demande peu d'interprétation à l'expert chargé d'une telle formalisation du jugement.

La limite principale de ce modèle est peut-être le manque d'expérimentations, qui justifieraient la théorie. On peut, en effet, se demander si celle-ci est suffisamment générale pour être appliquée à tous les types de décisions, de faits et de normes, bien qu'une expérimentation probante ne pourrait pas encore la prouver scientifiquement.

Le Professeur Wroblewski n'établit pas, par exemple, de liste complète des directives. Cela veut-il dire qu'il est impossible de les énumérer ? Ceci rendrait la théorie peu applicable.

Il serait sans doute très intéressant de comparer ce modèle avec d'autres modèles de la décision judiciaire, pour mieux le critiquer, mais rares sont ceux qui sont aussi riches et détaillés.

Section 4.7 : Extension du système d'aide à la décision judiciaire en fonction de la théorie de la décision juridique.

Dans quelle mesure le modèle théorique de la décision juridique peut-il être exploité dans une perspective de développement du prototype actuel ?

Une première utilisation serait de formaliser les jugements relatifs au cautionnement selon la structure proposée par Wroblewski, c'est-à-dire de repérer, pour chaque décision, les choix de validité, d'interprétation, de preuve et de conséquences.

Pour chacun d'eux, nous pourrions alors distinguer les faits, les normes, les directives et les évaluations, et tester la validité interne du raisonnement, en fonction du schéma proposé par le modèle.

Mais, comme le souligne le Professeur Wroblewski lui-même, une informatisation de ce test de cohérence ne peut être réalisée sans la définition d'échelles strictes de valeurs, qui semble irréalisable.

De plus ce travail de formalisation des jugements ne peut être demandé à chaque utilisateur désirant tester son propre projet de décision. Il devrait y consacrer trop d'efforts et de temps, avant d'aboutir à un résultat.

Une utilisation beaucoup plus réaliste du modèle théorique consiste à exploiter certaines notions, en les convertissant de telle sorte qu'elles soient compatibles avec notre propre représentation du raisonnement judiciaire.

A partir de la définition des faits et de leur classification donnée par le modèle, il serait intéressant d'étendre notre propre structure de faits. Ceux-ci sont actuellement repris dans la base de données sous forme de texte en langage naturel, ayant une polarisation et une pondération, un sens affirmé ou nié.

4.7.1 Introduction du temps et de l'espace.

La preuve d'un fait est définie comme

"Le fait F existe au temps t et au lieu p".

Les notions de temps et de lieu sont explicitement mises en évidence.

Comme nous l'avons déjà remarqué, elles ont un impact sur la structuration du jugement : sur la polarisation et la pondération des faits et standards, sur les compositions du jugement et sur le jugement lui-même.

L'introduction des coordonnées spatio-temporelles nous permettrait d'élargir le système de recherche documentaire dans les conditions déjà citées à la Section 2.1.

4.7.2 Typologie des faits.

Le Professeur Wroblewski classe les faits d'un jugement de trois manières : descriptifs ou évaluatifs, positifs ou négatifs, simples ou relationnels.

Il serait intéressant de comparer les concepts impliqués par ces classifications avec ceux que nous considérons actuellement.

Les faits positifs ou négatifs sont-ils nos faits affirmés ou niés?

Les faits évaluatifs se rapportent à une valeur, tandis que les faits négatifs ou relationnels sont définis par rapport à une norme. Retrouve-t-on ces relations dans notre schéma ?

A quoi correspond la notion de norme ? Peut-elle être considérée comme un standard, au sens où nous l'entendons ?

Que signifie la relation d'un fait à une norme dans le cas des faits relationnels ?

La réponse à ces questions permettra d'élargir la structure sémantique des faits et standards, et donc de donner de nouveaux critères de sélection et de nouveaux tests de jugements.

C'est cette optique que nous retenons comme extension possible du SIAD judiciaire en fonction de la théorie de la décision judiciaire du Professeur Wroblewski.

CHAPITRE 5 : Modèle empirique de l'argumentation juridique. [SCHA 86]

Section 5.1 : But poursuivi.

Le point de départ de cette analyse est la question suivante : Dans quelle situation se trouve l'avocat ou le juge qui désire défendre une cause et mener sa plaidoirie ?

Il prend en compte un certain nombre de faits du cas concret qui l'occupe et il cherche à les agencer de la meilleure façon pour convaincre le tribunal.

Il risque donc d'être intéressé par un système lui permettant d'identifier l'argumentation la plus persuasive, en fonction de faits introduits.

Une étape supplémentaire de réalisation de ce système lui offrirait la possibilité de tester la crédibilité de projets d'argumentations, et de filtrer les jugements introduits dans la base de données grâce aux mêmes critères.

L'intérêt du modèle est donc de guider la réalisation d'un prototype d'aide à l'élaboration d'argumentations.

Les critères d'efficacité d'une argumentation sont variables. Les facteurs qui semblent influencer le juge sont, à première vue, les suivants :

- 1) Les sources du droit sur lesquelles se fonde l'argumentation, en tenant compte de la hiérarchie des sources.
- 2) Un enchaînement cohérent des faits et de la thèse inférée.
- 3) La cohérence juridique de la motivation.
- 4) Le caractère vraisemblable des faits allégués par le plaideur.
- 5) La clarté des conclusions.
- 6) La réputation du plaideur.

Les facteurs 4,5 et 6 ne peuvent être pris en compte pour une informatisation.

Dans le système actuel, les tests de cohérence interne assurent partiellement les facteurs 2 et 3.

La prise en compte du point 1 constituerait un complément intéressant aux deux autres. L'objet du modèle empirique de l'argumentation juridique décrit dans les sections suivantes est d'atteindre cet objectif.

Section 5.2 : Dispositif d'une décision judiciaire.

Le dispositif d'une décision juridique est en principe le résultat de la mise en rapport d'éléments de faits avec une norme.

Avant d'être pris en considération par le juge, ceux-ci sont médiatisés par les organes sensoriels humains et par le langage.

Ceux qui sont sélectionnés par le juge sont ceux qui lui semblent établis et, de ces derniers, ne sont retenus que ceux qui lui paraissent pouvoir être mis en rapport avec une norme juridique, qui elle aussi a été sélectionnée parmi un éventail de règles.

La qualification du fait consiste donc en une spécification de la norme générale en fonction du cas d'espèce. Ce faisant, le juge a dégagé des conditions d'utilisation d'une norme juridique.

Section 5.3 : Contexte d'application d'un standard.

Il se peut qu'un juge voit la norme d'une autre manière qu'un autre juge et estime, à la différence de ce dernier, que la norme présuppose plus, moins ou d'autres conditions.

Il est toutefois probable que tous les juges considèrent que la norme présuppose entre autres, un même ensemble de conditions.

Celles-ci seront d'ailleurs le plus souvent exigées pour chaque norme par la loi, la doctrine ou la Cour de Cassation.

C'est cette conjonction de **conditions minimales** de faits plus ou moins qualifiés que nous appellerons par la suite **contexte nécessaire d'application** d'une norme, ou puisque nous avons pris l'habitude de considérer la norme dans un sens plus large, **d'un standard**.

Il est possible qu'un standard fasse l'objet d'un vide juridique, et ne soit contrôlé par aucune autorité supérieure. Nous dirons que ce standard ne possède pas de contexte nécessaire d'application.

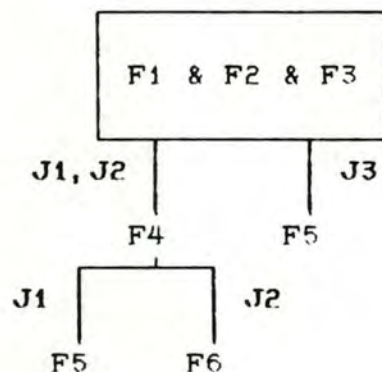
Par contre, plusieurs ensembles de conditions nécessaires d'application peuvent avoir été définis par la loi ou une autre source de droit pour un même standard. L'utilisation de l'un de ces ensembles seulement est nécessaire à l'application de ce dernier.

Les **conditions supplémentaires** exigées par telle ou telle juridiction pourraient être représentées par des parcours structurés en arborescences.

Dans l'exemple suivant, le standard S a été utilisé par les jugements J1, J2 et J3. La loi exige pour son application les conditions F1, F2 et F3. Le contexte nécessaire d'application du standard S est donc : F1 & F2 & F3.

Les juges de J1 et J2 ont considéré que le fait F4 était également une condition d'application du standard S ainsi que le fait F5 pour J1 et F6 pour J2. Le juge de J3 n'a pas jugé bon de retenir le fait F4, mais bien la condition F5.

Nous obtenons donc les parcours suivants :



Les chemins reliant des **conditions supplémentaires d'application du standard** reprises dans un même jugement sont appelées **contextes spécifiques d'application du standard**.

S'il était possible, pour chaque standard, de définir son contexte d'application, rassemblant ses contextes nécessaires et spécifiques, l'aide à l'argumentation se déroulerait comme suit :

Dans une première étape, le système rechercherait si les conditions minimales sont présentes dans le cas d'espèce.

Une seconde étape consisterait à vérifier si le cas d'espèce présente les conditions supplémentaires qu'une juridiction a sélectionnées pour appliquer le standard considéré.

Il resterait alors à choisir le parcours le plus intéressant parmi ces différentes combinaisons de conditions.

Section 5.4 : Méta-règles de l'argumentation basée sur la jurisprudence.

Pour déterminer le parcours sur lequel appuyer une argumentation la plus efficace possible, il est nécessaire de définir des **méta-règles** destinées à orienter la recherche du parcours le plus intéressant.

Il semble que les règles suivantes peuvent être envisagées :

1) Adopter le(s) standard(s) contrôlable(s) par la juridiction faisant le plus autorité.

Ainsi par exemple, en matière de cautionnement, il est plus intéressant de plaider la renonciation que la révocation car la première est contrôlable par la Cour de Cassation tandis que la seconde" fait l'objet d'une appréciation souveraine du juge de fond et n'est donc pas contrôlable par la juridiction suprême.

2) Orienter la recherche vers le parcours représentant la décision rendue par la juridiction ayant le plus de poids c'est-à-dire par celle qui fait le plus autorité.

3) S'il existe plusieurs parcours de même poids et qu'un ou plusieurs de ceux-ci représente(nt) une multiplicité de décisions, choisir le parcours en représentant le plus grand nombre.

4) S'il existe plusieurs parcours de même poids et s'il n'y a pas pluralité de décisions, choisir le parcours représentant la décision la plus récente.

La règle 3) ne peut être utilisée correctement qu'en regard de la hiérarchie du poids des décisions judiciaires, qu'il est bon de rappeler.

Les décisions judiciaires sont, dans le droit national belge, par ordre de poids décroissant :

1) Décisions de la Cour de Cassation statuant toutes chambres réunies, soit sur second pourvoi, soit dans l'intérêt de la loi.

2) Décisions de la Cour de Cassation statuant en audience plénière.

- 3) Décisions de la Cour de Cassation statuant sur premier pourvoi.
- 4) Décisions du juge du second degré statuant sur renvoi de la Cour de Cassation.
- 5) Décisions du juge du second degré.
- 6) Décisions du juge du premier degré.

Section 5.5 : Avantages et limites du modèle.

Le modèle de l'argumentation judiciaire est le produit d'une démarche empirique par laquelle le concepteur d'un SIAD juridique s'interroge sur les besoins réels de l'utilisateur juriste confronté à un litige.

En ce sens, le résultat est une approche très pragmatique du processus d'argumentation. Il permet de structurer logiquement et de manière simple les éléments intervenant dans la motivation d'une décision juridique.

Les seuls composants du modèle sont les standards et leurs contextes d'application, eux-mêmes composés de faits et de standards, ainsi que certaines méta-règles régissant l'utilisation optimale de ces contextes.

La qualité première de ce modèle est donc sa simplicité.

Un second avantage est sans aucun doute sa complémentarité avec la formalisation actuelle du jugement, puisqu'il reprend les faits, les standards et les jugements, en y ajoutant uniquement la notion de contexte d'application.

On pourrait également y retrouver la logique propositionnelle utilisée précédemment, si l'on considère, qu'un contexte de standard peut s'exprimer sous une forme logique.

Le concept de contexte d'application d'un standard signifie que la loi, ou le juge, établit que dès qu'un ensemble de faits ou de standards sont réunis, on peut leur appliquer une norme, ou

standard. Ces faits ou standards sont donc suffisants pour l'application de ce standard. La logique propositionnelle représente cette propriété sous forme d'une implication :

$$F_1, F_2, \dots, F_n \Rightarrow S$$

où les F_i sont les faits et standards conditions suffisantes d'application du standard S .

Plus particulièrement, le contexte nécessaire d'application d'un standard, comme son nom l'indique, est défini par la loi, la doctrine ou la Cour de Cassation, comme une conjonction de conditions de fait ou de droit nécessaires à l'application d'un standard. Cette relation de nécessité s'exprime également sous forme de logique propositionnelle dans l'implication :

$$F_1, F_2, \dots, F_n \Leftarrow S$$

où les F_i sont les faits et standards conditions nécessaires d'application du standard S .

Nous remarquons donc que le contexte nécessaire peut s'exprimer sous forme de l'équivalence logique :

$$F_1, F_2, \dots, F_n \Leftrightarrow S$$

où les F_i sont les faits et standards conditions nécessaires et suffisantes d'application du standard S .

Le plus gros inconvénient du modèle empirique est qu'il est lié à une analyse de la loi par un expert en vue d'y déceler tous les liens entre faits et normes, et qu'il est donc limité par l'expertise d'un juriste.

Le relevé des contextes d'application nécessite une interprétation objective des normes et des jugements, mais le modèle n'en donne aucune règle précise.

De plus, il se fonde sur l'hypothèse qu'il est toujours possible de définir les conditions nécessaires à l'application d'une norme, c'est-à-dire que "les juges considèrent que la norme présuppose entre autres un même ensemble de conditions" [SCHA 86]. Or, cette hypothèse n'est pas nécessairement toujours vérifiée pour toute norme.

De même, l'ordre de priorité des méta-règles, dans le cadre de la recherche du contexte le plus intéressant, peut être critiqué.

En résumé, ce modèle propose des solutions aisément envisageables dans le cadre du développement de notre SIAD judiciaire, mais entièrement dépendantes de la conception de la loi que se donne un expert.

Section 5.6 : Utilisation du modèle empirique dans le SIAD.

Le modèle de l'argumentation juridique tel qu'il vient d'être défini et critiqué, nous amène à proposer de nouvelles fonctions informatisées comme extensions du prorotype actuel :

La réalisation principale serait celle qui, en fonction d'un certain nombre de faits introduits et d'un objectif (polarisation) poursuivi par l'utilisateur, donnerait la liste des standards applicables à ces faits, éventuellement dans un ordre de préférence basé sur les méta-règles définies à la Section 5.4.

Le système serait également chargé d'anticiper l'argumentation de la partie adverse (dans le cas d'une plaidoirie ou d'un réquisitoire), en proposant une argumentation pouvant être dirigée sur base des mêmes faits, mais par l'adversaire. L'avocat pourrait alors contrer ce raisonnement en proposant une argumentation de poids plus important.

Le juriste pourrait aussi consulter le système dans le but de connaître, pour chaque standard, ses conditions d'application nécessaires et spécifiques.

Un test de validité supplémentaire viserait à exclure les décisions ou les projets de décisions dont l'argumentation ne contiendrait pas les prémisses suffisantes.

De même, les jugements cassés ou réformés pourraient servir de base à la définition de combinaisons de faits et standards à proscrire pour l'application de certaines normes.

Ces deux tests constitueraient un filtre lors de l'introduction des jugements dans la base de données.

Il serait également intéressant d'évaluer la décision rendue suite à la plaidoirie, de façon à comparer l'argumentation défendue et le résultat obtenu. Si celui-ci n'allait pas dans le

sens désiré, il conviendrait alors d'examiner la décision afin d'y déceler d'éventuelles incorrections dont l'analyse permettrait alors de soutenir une procédure d'appel.

Les applications informatiques de ce modèle sont donc nombreuses et semblent bien adaptées à l'objectif du SIAD qui est de constituer un support de décision, d'argumentation en l'occurrence.

CHAPITRE 6 : Comparaison des modèles théorique et empirique.

Section 6.1. : Justification interne d'un jugement.

Cette section a pour objectif de comparer les modèles théorique et empirique. Tout en étant apparemment très différents, ces deux modèles nous ont pourtant semblé posséder certains points communs, qu'il convient de mettre en évidence.

Tous deux tentent d'établir un schéma formel de la motivation de la décision judiciaire, et de définir ainsi des conditions de validité du raisonnement juridique.

Tandis que le Professeur Wroblewski décortique la prise de décision pour y trouver les décisions élémentaires de validité, d'interprétation, de preuve et de choix de conséquences, les chercheurs du C.R.I.D. isolent des liens logiques a priori entre les normes et leurs conditions d'application.

Pour Wroblewski, un jugement sera déclaré valide si l'enchaînement des faits, normes et directives respectent la règle du modus ponens : $((P \Rightarrow Q) \ \& \ P) \Rightarrow Q$. C'est ce qu'il appelle la justification interne de la prise de décision.

Pour le modèle empirique de l'argumentation, une décision judiciaire est correcte si chaque standard y est invoqué par l'intermédiaire de son contexte d'application nécessaire.

La notion de justification interne considérée dans le premier modèle joue donc le même rôle que celle de respect du contexte d'application des standards dans le second.

Ne pourrait-on pas aller plus loin dans la comparaison, en disant que la démarche nécessitée par la recherche des contextes d'application des normes utilise sans doute implicitement les règles énoncées par le Professeur Wroblewski ?

En effet, pour établir ces liens a priori entre éléments de fait et de droit, il faut, pour chaque norme, déterminer ses conditions de validité, l'interpréter, et en déduire une ou plusieurs règles de preuve.

Ces décisions successives sont évidemment très proches des trois premiers types de choix dont parle Wroblewski. L'expert juriste utilise vraisemblablement un ensemble de directives d'interprétation du 1er ou du 2d degré pour aboutir à une concrétisation de la loi sous forme de liens de contextes.

Certains contextes sont en fait eux-mêmes des directives de preuve, puisqu'ils établissent une règle d'implication entre les faits observés et leur conséquence juridique, qui est le fait qualifié.

D'autres établissent un lien entre ces faits qualifiés et la norme qui leur est applicable.

Les hypothèses et les règles préalables à la modélisation sous forme de contextes d'argumentation pourraient être celles du modèle théorique, qui couvre l'entièreté de la démarche du juriste, depuis le choix de la norme jusqu'à celui des conséquences.

Le modèle empirique ne détaille pas la technique d'interprétation de la loi et en laisse la liberté à l'expert. C'est un modèle partiel de la décision judiciaire.

Pourtant, ce dernier possède une dimension non prépondérante dans le modèle théorique : la référence à la jurisprudence. Elle est selon lui un élément indispensable au développement d'une argumentation convaincante. En effet, l'avocat rendra sa plaidoirie plus crédible encore s'il peut faire constater par le juge que les circonstances de fait qu'il avance ont servi à défendre une thèse équivalente dans un jugement précédemment rendu.

Le modèle de l'argumentation n'est donc pas basé uniquement sur la loi ou la doctrine, mais également sur le mécanisme du "précédent".

En conclusion, nous pouvons affirmer que les deux modèles ont en commun la volonté de définir des règles de validité d'une argumentation, basée sur une analyse de la loi, mieux détaillée chez Wroblewski, mais complétée par un relevé de la jurisprudence, dans le modèle empirique.

Section 6.2 : Choix du modèle empirique.

Pour poursuivre l'élaboration du prototype de SIAD judiciaire, nous avons choisi d'exploiter le second modèle étudié, celui de l'argumentation.

Les raisons de ce choix sont, comme cela a déjà été dit à la Section 5.5, la simplicité de ses termes et surtout son adéquation au modèle de raisonnement du système actuel, ainsi qu'à la structure de la base de données de jugements.

Le modèle théorique aurait nécessité un trop gros effort de détermination des composants de la décision : directives, critères d'évaluation, etc. Ces éléments sont même parfois impossibles à repérer, telles les échelles de valeurs dont il a déjà été question.

De même, l'informatisation de ce modèle donnerait un cadre trop rigide au processus de décision, et éliminerait toute forme d'intuition et de liberté du juge. Comment pourrait-on en effet justifier un jugement rendu sur base d'une intime conviction, par exemple ?

Il ne paraissait pas non plus utile aux juristes consultés de revoir immédiatement la typologie et la structuration des faits en fonction du temps, comme cela avait été proposé à la Section 4.7, tandis que la perspective d'une aide à l'argumentation sur base du modèle empirique leur semblait d'un intérêt pratique plus important.

Celui-ci, par son aspect concret, nous autorise à imaginer très rapidement de nouvelles fonctions d'aide à l'élaboration d'argumentations.

Malgré ses limites, en particulier celle qui le rend dépendant de l'expert, et qu'il convient de garder à l'esprit, c'est ce modèle qui sera développé dans les chapitres suivants.

CHAPITRE 7 : Extension du modèle d'aide à la décision judiciaire: Modèle d'aide à l'argumentation.

Ce chapitre sera consacré à la représentation conceptuelle du modèle de l'argumentation juridique, qui servira de base à l'extension du prototype comme système d'aide à l'argumentation.

Section 7.1. : Objectif du système.

Avant de formaliser les concepts du modèle, il est nécessaire de choisir l'optique d'utilisation qui sera donnée à l'extension du prototype.

L'objectif du développement est d'aider le juriste à élaborer une argumentation qui soit correcte et efficace.

Le système s'adresse à un avocat ou un juge disposant de faits du cas d'espèce à traiter, afin de lui conseiller un certain nombre de standards, ou normes, à retenir dans sa motivation.

Pour l'avocat, il s'agit de choisir les standards qui auront le plus de poids sur la décision dans l'intérêt de son client. Pour le juge, le système vérifiera si toutes les conditions sont réunies pour l'utilisation du ou des standards qu'il désire invoquer.

Le juge peut également être intéressé par le relevé des décisions de la Cour de Cassation portant sur une utilisation incorrecte ou incomplète de la loi, de façon à éviter la reproduction de telles erreurs et le risque de casse du jugement.

Le système devrait aussi permettre à son utilisateur de mettre à jour régulièrement les contextes de standards, en conservant pour chacun leur date et leur origine (loi, doctrine, cassation).

Une fois cet objectif défini, nous pouvons maintenant repérer les notions-clés du modèle qui nous seront utiles par la suite pour la création des fonctions.

Section 7.2. : Définition des concepts.

Le concept central du modèle est le **contexte d'application d'un standard**, composé de ses **contextes d'application nécessaires et spécifiques**.

Le **contexte nécessaire d'application d'un standard** est un ensemble de faits qualifiés et/ou de standards nécessaires à l'application du standard considéré, par définition de la loi, de la doctrine ou de la Cour de Cassation.

Ces faits et standards seront appelés **conditions nécessaires d'application du standard**.

Le **contexte spécifique d'application d'un standard** est un ensemble de faits ou standards ayant servi d'argumentation au standard considéré dans la jurisprudence, en supplément d'un contexte nécessaire.

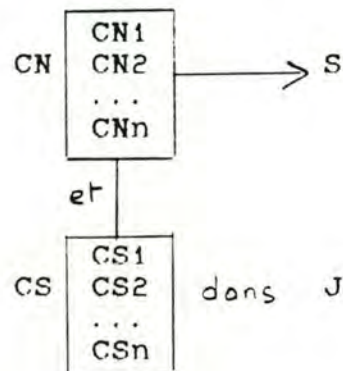
Un contexte spécifique s'associe toujours à un contexte nécessaire, sauf lorsque le standard fait l'objet d'un vide juridique, c'est-à-dire qu'aucun contexte nécessaire n'a été défini pour son application.

Ces faits et standards seront appelés **conditions spécifiques d'application du standard**.

Les valeurs de ces contextes peuvent évoluer, par décision de la Cour de Cassation, ou vote d'une loi y dérogeant.

De même, un standard peut posséder plusieurs contextes nécessaires, pour autant que les autorités aient défini une alternative parmi ses conditions d'application.

Nous pourrions représenter les concepts énumérés ci-dessus de façon graphique :



ou sous forme de logique propositionnelle :

$$S \Leftrightarrow (CN1 \ \& \ CN2 \ \& \ \dots \ \& \ CNn)$$

$$\text{Dans J: } [(CN1 \ \& \ CN2 \ \& \ \dots \ \& \ CNn) \ \& \ (CS1 \ \& \ CS2 \ \& \ \dots \ \& \ CSn)] \Rightarrow S$$

Ces deux représentations signifient que le standard S a pour contexte d'application nécessaire, et suffisant, CN, conjonction des conditions nécessaires CN1, CN2, ..., CNn, et que le contexte spécifique CS formé des conditions spécifiques CS1, CS2, ..., CSn, est associé au contexte CN dans le jugement J.

Une des conditions composant ces contextes peut être elle-même un standard auquel on attribue également 0, 1 ou plusieurs contextes nécessaires et spécifiques.

Nous appellerons **contexte de non application d'un standard** un ensemble de conditions de faits et/ou de droit ayant fait l'objet d'un refus par la Cour de Cassation, et éventuellement par la doctrine ou la loi, pour l'application de ce standard.

Cette notion pourrait également être formalisée sous forme de logique propositionnelle :

$$\sim [S \Leftrightarrow (CN1 \ \& \ CN2 \ \& \ \dots \ \& \ CNn)].$$

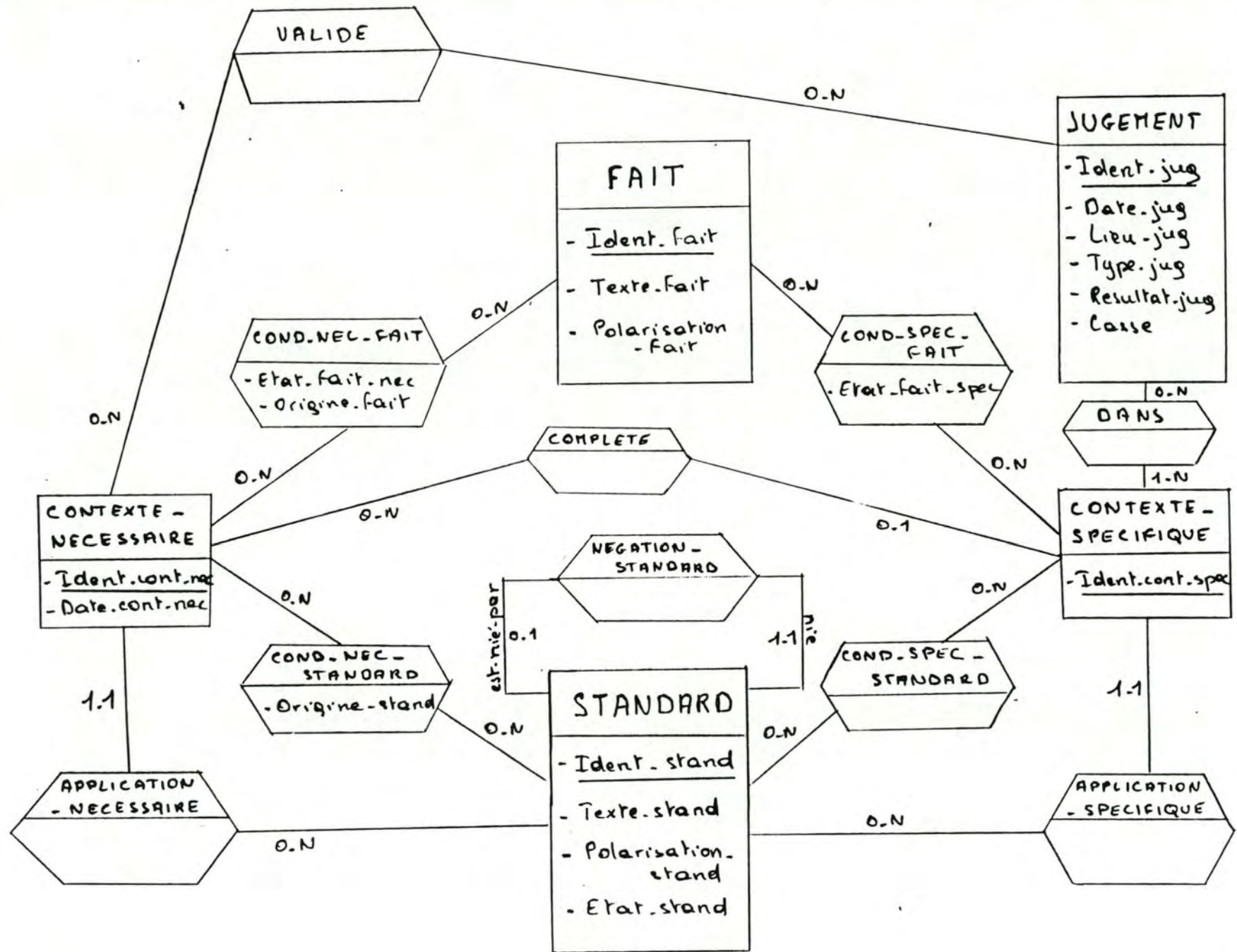
Cette formule signifie que le standard S n'est pas applicable grâce aux conditions CN1 à CNn.

Munis de ces quelques concepts, auxquels nous ajouterons ceux de faits, standards et jugements, définis dans le schéma de la base de jugements, nous possédons tous les éléments nécessaires au développement des fonctions d'aide à l'argumentation.

Section 7.3. : Schéma conceptuel des données.

Ces considérations nous amènent à établir un schéma conceptuel des données déjà utilisé pour représenter le jugement dans le prototype actuel.

SCHEMA CONCEPTUEL DES DONNEES :



DESCRIPTION DU SCHEMA CONCEPTUEL :

Les types d'entités :

JUGEMENT :

Un jugement est une décision juridique relative au problème de cautionnement restreint au cas nous intéressant. [HICK 82]

- identifié par "Ident-jugement"
ou par "date" et "lieu"
- résultat (libération ou résultat) donné par "résultat"
- instance donnée par "type"
- "casse" est un booléen signifiant si le jugement a été cassé ou non.

FAIT :

Un fait est fait qualifié pouvant intervenir dans un raisonnement juridique.

- identifié par "ident-fait"
- polarisation donnée par "polarisation-fait"
- texte affirmé donné par "texte-fait"
- texte nié donné par "texte-négation".

STANDARD :

Un standard est un élément de droit pouvant intervenir dans un raisonnement juridique.

- identifié par "ident-standard"
- texte donné par "texte-standard"
- polarisation donnée par "polarisation-standard"
- état affirmé ou nié donné par "état-standard".

CONTEXTE-NECESSAIRE :

Un contexte nécessaire est une conjonction de conditions nécessaires à l'application d'un standard et définies par la loi, la doctrine ou la Cour de Cassation.

- identifié par "Ident-cont-nec"
- date donnée par "date-cont-nec".

CONTEXTE-SPECIFIQUE :

Un contexte spécifique est une conjonction de conditions faits ou standards, ayant servi dans un ou plusieurs jugements d'argumentation pour l'application d'un standard en plus d'un contexte nécessaire.

- identifié par "ident-cont-spec".

Les types d'associations :

NEGATION-STANDARD (STANDARD, STANDARD) :

correspondance entre un standard à l'état affirmé et son correspondant nié.

APPLICATION-NECESSAIRE (STANDARD , CONTEXTE-NECESSAIRE) :

relation entre un standard et un de ses contextes d'application nécessaires.

COND-NEC-STANDARD (CONTEXTE-NECESSAIRE , STANDARD) :

Présence d'un standard dans le contexte d'application nécessaire d'un autre standard.

- "origine-stand" : origine de l'expression du standard comme condition nécessaire (loi, doctrine, cassation).

COND-NEC-FAIT (CONTEXTE-NECESSAIRE , FAIT) :

Présence d'un fait qualifié dans le contexte d'application nécessaire d'un standard.

- "origine-fait" : origine de l'expression du fait qualifié comme condition nécessaire du standard (loi, doctrine, cassation).

- "état-fait-nec" : état affirmé ou nié du fait comme condition nécessaire d'application du standard.

APPLICATION-SPECIFIQUE (STANDARD , CONTEXTE-SPECIFIQUE) :

- relation entre un standard et un de ses contextes d'application spécifiques.

COND-SPEC-STANDARD (CONTEXTE-SPECIFIQUE , STANDARD) :

Présence d'un standard dans le contexte d'application spécifique d'un autre standard.

COND-SPEC-FAIT (CONTEXTE-SPECIFIQUE , FAIT) :

Présence d'un fait qualifié dans le contexte d'application spécifique d'un standard.

- "état-fait-spec" : état affirmé ou nié du fait comme condition spécifique d'application du standard.

DANS (CONTEXTE-SPECIFIQUE , JUGEMENT) :

Présence dans un jugement d'un contexte d'application spécifique d'un standard.

COMPLETE (CONTEXTE-SPECIFIQUE , CONTEXTE-NECESSAIRE) :

Correspondance entre un contexte spécifique et le contexte nécessaire qu'il complète ou précise.

VALIDATION (CONTEXTE-NECESSAIRE , JUGEMENT) :

Présence dans un jugement d'un contexte d'application nécessaire d'un standard.

CONTRAINTES D'INTEGRITE :

- Un contexte nécessaire est toujours associé soit à un fait au moins, soit à un standard au moins, par les associations cond-nec-fait ou cond-nec-stand.
- Un contexte-spécifique est toujours associé soit à un fait au moins, soit à un standard au moins, par les associations cond-spec-fait ou cond-spec-stand.
- Un standard ne peut être une condition nécessaire ou spécifique de son propre contexte d'application, ou de celui d'un standard qui est lui-même une de ses conditions d'application à un degré quelconque.
- A un standard et un jugement donnés correspond un seul contexte spécifique.

- Si un contexte spécifique d'un standard n'est relié à aucun contexte nécessaire de ce standard, c'est qu'il n'existe aucun contexte nécessaire pour ce standard.
- Les différentes occurrences de contextes spécifiques associés à un même contexte nécessaire apparaissent dans des jugements distincts.
- Un contexte nécessaire associé à un ou plusieurs contextes spécifiques doit être relié par l'association validation au moins aux jugements dans lesquels apparaissent ces contextes spécifiques.

COMMENTAIRES :

Quelques remarques s'imposent ici pour la bonne compréhension du modèle.

Le type d'entité JUGEMENT correspond à celui du même nom défini à la section 1.2, auquel nous avons ajouté l'attribut "casse", de façon à trier les jugements (d'instance uniquement) en ne considérant pour la gestion des contenus que ceux qui n'ont pas été réputés invalides par la Cour de Cassation. Le domaine de valeurs de cet attribut est l'ensemble {oui, non}. L'attribut "instance" prend ses valeurs dans l'ensemble {CASSATION TOUTES CHAMBRES REUNIES, CASSATION SEANCE PLEINIERE, CASSATION SECOND RENVOI, CASSATION, SECOND DEGRE, PREMIER DEGRE}.

Ce schéma conceptuel, contrairement au précédent, distingue les types d'entité FAIT et STANDARD, car un standard, à l'opposé d'un fait, est maintenant associé à un contexte nécessaire, par l'association APPLICATION-NECESSAIRE et à un contexte spécifique par l'association APPLICATION-SPECIFIQUE. Plutôt que de gérer une contrainte d'intégrité encombrante, nous avons préféré dissocier ces deux concepts qui n'ont plus le même ensemble de propriétés.

De plus, il nous a paru utile de représenter par des occurrences différentes le standard à l'état affirmé et ce même standard à l'état nié. Ceci nous permettra de gérer indépendamment les contextes nécessaires et spécifiques d'application et de non application de standards, ces derniers étant représentés comme les premiers, mais étant, eux, associés au standard nié. L'association NEGATION-STANDARD a été ajoutée pour gérer la correspondance entre les deux états d'un même standard.

Tous les standards affirmés n'ont pas nécessairement de correspondant nié, soit parce que leur négation n'a aucun sens, soit parcequ'elle n'a jamais été considérée dans la loi ou la jurisprudence. Ceci est exprimé par la connectivité 0-1 du rôle "est-nié-par" de l'association NEGATION-STANDARD.

Le concept de pondération de fait ou standard a disparu de la représentation du raisonnement, pour la raison suivante : nous considérons que tous les faits et standards repris comme conditions nécessaires ou spécifiques sont indispensables au respect du contexte d'application qu'ils composent, et qu'ils sont donc tous "importants". Autrement dit, l'absence d'une seule condition d'application d'un standard, comme prémisses de ce standard, rend l'argumentation invalide. On ne pourra pas dire qu'une condition est plus importante qu'une autre dans un contexte. Il faudra bien entendu tenir compte de cette remarque lors de la description des contextes nécessaires et spécifiques de chaque standard.

La polarisation des faits et standards conserve le même domaine de valeurs {libération, maintien}. Nous continuons donc à traiter le problème juridique du cautionnement, bien que le modèle soit à première vue applicable à n'importe quel type de normes.

Lorsque plusieurs possibilités d'application ont été décrétées pour un standard, elles sont désignées par des occurrences distinctes du type d'entité CONTEXTE-NECESSAIRE.

Nous décidons de conserver la date à laquelle a été défini chaque contexte d'application nécessaire, car elle peut constituer un élément supplémentaire d'appréciation pour le juriste qui désire effectuer un choix parmi les contextes d'un même standard. L'origine de chaque condition nécessaire est également retenue, pour la même raison.

Les faits, standards et jugements peuvent être représentés sans pour cela être associés à des contextes de conditions. Nous obtenons alors un schéma suffisamment général pour pouvoir être rapporté au modèle précédent, et être éventuellement étendu par la suite.

CHAPITRE 8 : Fonctions d'aide à l'argumentation juridique.

Grâce à ce modèle, nous pouvons maintenant spécifier et implémenter les fonctions informatiques d'aide à l'argumentation. Les sections suivantes ont pour but de décrire l'extension apportée au prototype de SIAD judiciaire.

Section 8.1. : Spécification des fonctions.

8.1.1. Fonctions d'aide à l'argumentation.

1.. RECSTAND

But : Donner les standards ayant un effet juridique déterminé et leurs conditions d'application.

Entrée : **p** : Polarisation ("+" pour libération, "-" pour maintien).

Résultat : Standards de polarisation **p** et leurs contextes d'application nécessaires et spécifiques. Les faits, standards et contextes sont représentés par leur numéro identifiant.

A la demande de l'utilisateur, les contextes seront affichés dans un ordre de priorité régi par les méta-règles définies à la Section 5.4. et interprétées dans le programme par les critères suivants :

~ Les standards ayant un contexte d'application nécessaire.

- Les contextes sont ensuite classés dans l'ordre suivant, le critère i+1 étant applicable pour départager les ex-aequo après application du critère i :

1. Contextes nécessaires validés par un jugement de Cassation.
2. Contextes nécessaires dont une condition au moins a pour origine la loi.
3. Contextes validés par le plus grand nombre de jugements ayant eu l'effet désiré.
4. Contextes validés par le plus grand nombre de jugements de Cassation Toutes Chambres Réunies ayant eu l'effet désiré.
5. Contextes validés par le plus grand nombre de jugements de Cassation Séance Plénière ayant eu l'effet désiré.
6. Contextes validés par le plus grand nombre de jugements de Cassation Second Pourvoi ayant eu l'effet désiré.
7. Contextes validés par le plus grand nombre de jugements de Cassation ayant eu l'effet désiré.
8. Contextes validés par le plus grand nombre de jugements du Second Degré ayant eu l'effet désiré.
9. Contextes validés par le plus grand nombre de jugements du Premier Degré ayant eu l'effet désiré.
10. Contextes rendus le plus récemment par la juridiction la plus haute.

Un exemple d'exécution de cette fonction RECSTAND se trouve en annexe.

2.. STANAIDE.

But : Donner les standards applicables à des faits et un effet juridique déterminés.

Entrées : (p : polarisation ("+" ou "-")).

F : Liste de numéros identifiants de faits.

Résultat : Standards (de polarisation p) applicables grâce aux faits de la liste F et leurs contextes d'application, éventuellement dans un ordre de priorité déterminé par les critères décrits pour la fonction RECSTAND.

Si l'utilisateur le souhaite, le système propose les contextes qui seraient également applicables s'il parvenait à prouver un seul fait supplémentaire quelconque.

Vous trouverez en annexe un exemple d'exécution de la fonction STANAIDE.

8.1.2. Fonctions de consultation.

RECCONT.

But : Donner les contextes d'application d'un standard.

Entrée : s : numéro identifiant d'un standard.

Résultat : Contextes nécessaires et spécifiques associés à ce standard (numéro, date, jugements associés).

8.1.3. Fonctions de mise à jour.

MAJCONT.

But : Mise à jour des contextes d'application nécessaires et spécifiques des standards.

1. Ajout d'un contexte nécessaire.

Entrées : s : numéro de standard.

F : Liste de triplets (numéro de fait, état, origine : loi, doctrine ou cassation).

S : Liste de couples (numéro de standard, origine).

d : date.

Résultat : Ajout d'un contexte nécessaire de date **d** au standard **s**, dont les conditions d'application sont les faits d'état donné et les standards, d'origine donnée, dans les listes **F** et **S**.

Si le standard **s** n'avait aucun contexte d'application nécessaire, les contextes spécifiques qui y sont éventuellement associés sont supprimés ou corrigés en fonction des conditions nécessaires introduites :

Si un contexte spécifique possède toutes les conditions nécessaires, celles-ci lui sont supprimées et il est associé au nouveau contexte nécessaire.

Sinon, il est supprimé.

2.. Ajout d'un contexte spécifique.

Entrées : **s** : numéro de standard.
 cn : numéro de contexte nécessaire.
 F : liste de couples (numéro de fait, état).
 S : liste de numéros de standards.
 J : liste de numéros de jugements.

Résultat : Ajout d'un contexte spécifique au standard **s** dont les conditions d'application sont les faits d'état donné et les standards des listes **F** et **S**. Ce contexte est associé au contexte nécessaire **cn** et aux jugements de la liste **J**.

3.. Retrait d'un contexte nécessaire.

Entrées : **s** : numéro de standard.
 cn : numéro de contexte nécessaire.

Résultat : Suppression du contexte nécessaire **cn** d'application du standard **s**.

Si ce contexte était associé à un contexte spécifique, celui-ci est supprimé.

4.. Retrait d'un contexte spécifique.

Entrées : **s** : numéro de standard.
 cs : numéro de contexte spécifique.

Résultat : Suppression du contexte spécifique **cs** d'application du standard **s**.

5.. Modification d'un contexte nécessaire.

5.1.. Ajout de faits ou standards.

Entrées : **s** : numéro de standard.

cn : numéro de contexte nécessaire.

F : liste de triplets (numéro de fait, état, origine)
ou de couples (numéro de standard, origine).

Résultat : Ajout des faits d'état donné ou de standards, d'origine donnée, comme conditions nécessaires d'application dans le contexte nécessaire **cn** du standard **s**.

Les contextes spécifiques associés au contexte nécessaire modifié sont corrigés ou supprimés en fonction des conditions nécessaires introduites :

Ceux qui contiennent les faits et standards ajoutés seront simplement amputés de ceux-ci, tandis que les autres seront supprimés.

5.2.. Suppression de faits ou standards.

Entrées : **s** : numéro de standard.

cn : numéro de contexte nécessaire.

F : liste de numéros de faits ou de standards.

Résultat : Suppression des faits ou standards de la liste **F** comme conditions nécessaires d'application du contexte nécessaire **cn** du standard **s**.

Les faits ou standards seront ajoutés comme conditions des contextes spécifiques associés au contexte nécessaire modifié.

Si toutes les conditions du contexte nécessaire **cn** sont supprimées et qu'il existe un autre contexte nécessaire pour ce standard, tous les contextes spécifiques associés au contexte nécessaire supprimé sont supprimés.

6.. Modification d'un contexte spécifique.

6.1.. Ajout de faits ou standards.

Entrées : **s** : numéro de standard.

cs : numéro de contexte spécifique.

F : liste de couples (numéro de fait, état)
ou de numéros de standards.

Résultat : Ajout des faits d'état donné ou de standards comme conditions spécifiques d'application dans le contexte spécifique **cs** du standard **s**.

6.2. Suppression de faits ou standards.

Entrées : **s** : numéro de standard.
 cn : numéro de contexte spécifique.
 F : liste de numéros de faits ou de standards.

Résultat : Suppression des faits ou standards de la liste **F** comme conditions spécifiques d'application du contexte spécifique **cs** du standard **s**.

Section 8.2. : Implémentation des fonctions.

Le prototype de SIAD judiciaire a été jusqu'à présent développé sur un ordinateur de type PC, ce qui se justifie aisément par la plus grande accessibilité qu'offre ce genre de machine pour un avocat ou un juge.

L'extension vers un prototype d'aide à l'argumentation réalisant les fonctions précédemment décrites se poursuivra donc dans cette optique, de façon à garder une certaine uniformité dans la conception.

Le choix de l'IBM PC et du système d'exploitation MS-DOS étant ainsi confirmé, il restait à décider d'un langage de programmation approprié.

Le langage DBASE2 a été abandonné après avoir démontré sa complexité de manipulation et ses faiblesses de performances.

Par contre, le langage Prolog, ici le système Micro-Prolog, déjà employé pour traiter le test de validité externe, a été retenu pour deux raisons essentielles :

-

La première est sans doute la forme de raisonnement qui le caractérise, celle de l'implication ou du SI..., ALORS... . Nous avons déjà remarqué que le concept de contexte d'application d'un standard pouvait s'exprimer sous cette forme logique classique (Section 7.2.).

La recherche du meilleur standard à argumenter est en fait l'application du principe de résolution matérialisé par Prolog, puisqu'il suffit de vérifier si toutes les conditions sont bien remplies pour l'application d'un standard, auquel cas celui-ci est retenu pour l'argumentation.

La particularité du modèle des contextes est d'être récursif, puisqu'un standard condition d'application d'un autre standard possède également son propre contexte d'application, et ainsi de suite. Ce phénomène est tout à fait traitable par Prolog, dont une propriété importante est sa capacité de gérer la récursivité.

La seconde raison est la forme même des faits Prolog, représentés par des prédicats du premier ordre dont le calcul couvre celui de l'algèbre relationnelle, exploitée dans la base de données de jugements du précédent prototype.

De cette façon, les faits Prolog sont une traduction immédiate des relations dérivées du modèle E/A grâce aux principes de transformation de schémas déjà utilisés sur le modèle du jugement (Section 1.2.).

Le développement d'une interface entre le prototype actuel et son extension sera donc facilité par cette équivalence de représentation.

Prolog nous paraît de plus bien adapté à la manipulation de listes de longueur variable et illimitée, comme les listes de faits représentant le cas étudié, les listes de faits et standards conditions d'application d'un standard ou les listes de jugements spécifiant un standard.

L'ensemble de ces raisons nous ont poussés à choisir Prolog comme langage de programmation pour les fonctions spécifiques d'aide à l'argumentation, mais aussi les fonctions de mise à jour de contextes d'application de standards.

TROISIEME PARTIE :

PERSPECTIVES D'EVOLUTION.

Cette troisième partie a pour but d'effectuer une critique du système obtenu (chapitre 9) et de la représentation conceptuelle des données sous forme du schéma Entité/Association (chapitre 10). Le chapitre 11 analyse les nouveaux modes de représentation de connaissances dans le but de choisir, dans le chapitre 12, un nouveau formalisme de représentation des données et des traitements.

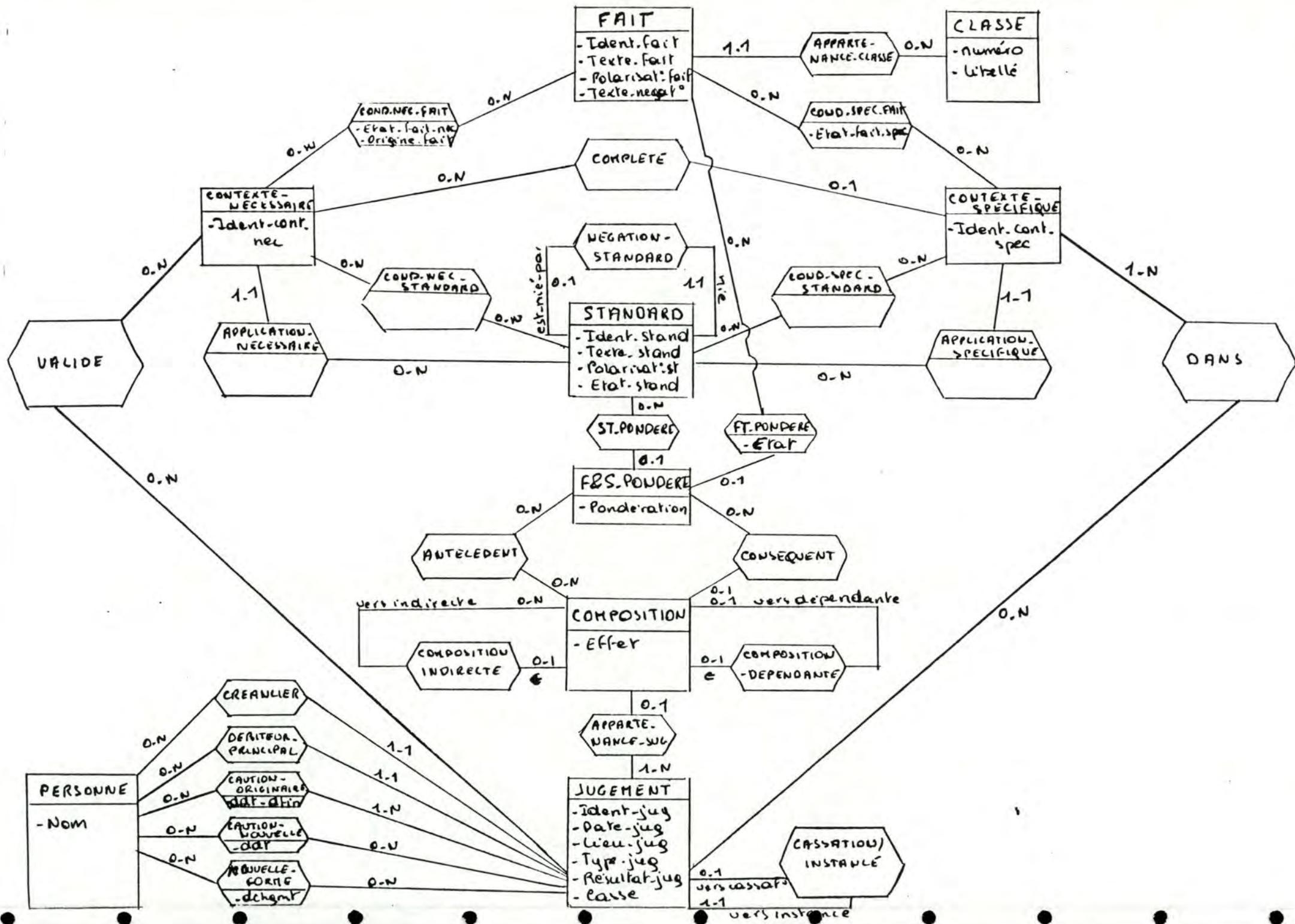
Le système obtenu après réalisation des fonctions d'aide à l'argumentation, bien que présentant les spécifications requises par le modèle de l'argumentation, pourrait être sensiblement amélioré en tenant compte des critiques suivantes : l'absence d'interface entre les deux systèmes, le manque de convivialité, les limites d'ordre sémantique, et le manque d'expérimentation.

La plus importante est l'absence d'interface commune au système d'aide à l'argumentation et au prototype de recherche documentaire et de tests de validité.

Etant réalisés dans des langages différents, ils sont appelés par l'utilisateur de façon indépendante. Les bases de données sont dédoublées, l'une sous forme de système relationnel, l'autre sous forme de faits Prolog. Une modification dans l'une n'est donc pas répercutée immédiatement dans l'autre.

Il suffirait pourtant de très peu de modifications sur le schéma conceptuel du jugement pour y intégrer le schéma conceptuel des contextes : division de l'entité FAIT&STANDARD en deux entités, FAIT et STANDARD, cette dernière reprenant les occurrences des standards affirmés et niés et ajout de l'attribut "casse" au type d'entité jugement.

L'intégration des deux schémas peut en effet donner le schéma suivant :



Une interface devrait donc être réalisée entre les différents éléments du système, de façon à concilier les avantages de Prolog (inférences, implications, récursivité) et ceux d'un SGBD relationnel (accès à un grand nombre de données organisées, algèbre relationnelle).

Nous obtiendrions donc un système unique permettant à la fois une recherche documentaire sur la base de données de jugements, des conseils sur la meilleure argumentation à défendre en fonction du cas traité et une évaluation de jugements présents dans la base ou de propositions de décisions.

Un deuxième inconvénient du prototype d'aide à l'élaboration d'argumentation est son manque de convivialité, dû aux caractéristiques de sorties à l'écran du langage Micro-Prolog.

Il est par exemple impossible de désigner un point de l'écran par ses coordonnées et d'y faire apparaître un caractère. La gestion d'écran s'effectue donc ligne par ligne, l'ensemble des lignes remontant chaque fois qu'un nouveau caractère de fin de ligne est introduit.

C'est pourquoi il serait utile de rechercher un autre environnement que celui de Micro-Prolog, qui offrirait une gestion d'écran agréable pour l'utilisateur, intégrée à un langage de type Prolog.

Une étape supplémentaire dans l'amélioration de l'interface utilisateur du prototype pourrait être la création d'un langage permettant au juriste d'introduire les faits, non pas par leur numéro identifiant, mais bien par un ensemble de mots-clés, qui désigneraient de façon univoque les faits connus dans la base de données.

De même, les standards pourraient être désignés par un mot représentatif de leur signification (exemple : standard de la "cause"), plutôt que par leur numéro.

Les mêmes termes seraient utilisés pour la représentation des contextes d'application de standards, puisqu'ils sont eux-mêmes composés de faits et de standards.

Une révision de la structure sémantique des faits et standards permettrait de découvrir ces mots-clés.

Ces dernières considérations traitent des limites techniques du prototype réalisé. D'autres limites d'ordre sémantique peuvent être dégagées et donner lieu à des propositions d'extensions futures.

A la section 5.6., nous proposons déjà la possibilité D'un test portant sur l'introduction de jugements dans la BD. Celle-ci n'étant encore représentée que sous forme de faits Prolog, sans accès direct à la base de jugements d'origine, il nous a paru plus sage d'attendre que cette interface entre les deux types de systèmes soit créée pour développer cette nouvelle fonction.

En particulier, l'évaluation de la décision rendue suite à la plaidoirie pourrait faire l'objet de ce test de validité.

Mais la limite principale du système est son manque d'expérimentation sur des standards relatifs au cautionnement. Son utilité réelle ne pourra être vérifiée que lorsque des expériences auront été réalisées dans des cas concrets.

En conclusion, le développement d'une interface utilisateur commune aux différents types de systèmes et son expérimentation dans des conditions réelles d'exploitation sont les deux extensions à envisager en priorité.

La première nécessite la révision de la puissance sémantique du modèle Entité/Association dans l'expression des données relatives aux deux systèmes, c'est-à-dire aux jugements et aux contextes d'application, en le comparant à d'autres formalismes de représentation des connaissances.

Les chapitres suivants tentent de proposer un autre mode de représentation des connaissances que ceux du modèle E/A et de la logique propositionnelle, qui permettrait de revoir la conception des données et des traitements, dans le but de leur donner un formalisme commun de représentation.

CHAPITRE 2 : Critique de la représentation conceptuelle des données.

Le modèle Entité/Association a jusqu'à présent été utilisé pour la représentation des connaissances juridiques dans le cadre du SIAD judiciaire. Or, les SIAD ne sont pas, en général,

développés à partir de ce formalisme de représentation des connaissances : d'autres formalismes ont été créés expressément pour ce type de systèmes, et nous en parlerons dans le chapitre suivant.

Nous allons l'évaluer selon les critères suivants, qui nous ont paru importants pour l'appréciation d'un formalisme de représentation des connaissances : simplicité du formalisme, visualisation et communication avec l'utilisateur, capacité de structuration des informations, expression de relations n-aires, extensibilité, représentation des traitements, des contraintes d'intégrité, des valeurs par défaut, des cas d'exception, du texte et du temps, des occurrences, possibilité de raisonner directement sur le schéma de façon performante.

Le schéma Entité/Association présente, tout d'abord, de nombreux avantages pour la modélisation des données :

Il offre la possibilité de formaliser les relations de classification (regroupement d'occurrences de même type), d'agrégation (entités, associations, attributs d'entités, attributs d'associations), de regroupement (Associations de sous-ensemble sans héritage de propriétés), et même aujourd'hui de spécialisation (sous-typage avec héritage de propriétés), qui sont les mécanismes généraux d'abstraction à la base des processus de modélisation des connaissances. Il est donc et général capable de formaliser tous les types de connaissances, données ou modèles eux-mêmes.

Simplicité du formalisme : Ses composants sont peu nombreux, faciles à visualiser et à interpréter. Son apprentissage peut donc se faire relativement vite.

Visualisation et communication avec l'utilisateur : Bien manipulé, il conserve le mieux possible les termes des experts du domaine étudié. Il peut donc servir facilement d'outil de communication entre le concepteur et l'utilisateur un peu averti.

Mais il arrive qu'il donne une image tout à fait statique et rigide de la connaissance, qui peut parfois paraître artificielle. Dans le cas qui nous occupe, nous pouvons citer l'exemple de la "composition" dans un jugement, qui est en fait une implication logique, mais dont la représentation sous forme d'entité associée à l'entité FAIT&STANDARD-PONDERE par deux associations ANTECEDENT et CONSEQUENT, efface un peu l'aspect naturel de la logique.

Capacité de structuration des informations : Toutes les informations relatives à un concept sont rassemblées autour de l'entité qui le représente, soit par des attributs, soit par des associations. Elles sont donc directement visualisables.

Expression de relations n-aires (associations de degré n) : Les relations plus que binaires y sont également autorisées. Par conséquent, le modèle paraît un mode de représentation approprié des connaissances relationnelles.

Extensibilité : Comme nous avons pu le constater lors de l'extension du prototype d'aide à l'élaboration de jugements en un système d'aide à l'argumentation, un schéma est aisément extensible, et le schéma obtenu peut être décomposé en sous-schémas indépendants.

Si nous pensons à chercher un autre modèle de représentation des connaissances, c'est que le schéma Entité/Association utilisé jusqu'à présent nous a paru présenter quelques défauts par rapport aux usages que nous voulons faire des concepts retenus.

Certains types de connaissances sont assez malaisés à formaliser.

Les traitements : Le modèle est conçu pour représenter uniquement les données. Les traitements font l'objet d'un modèle distinct.

Les contraintes d'intégrité : Les contraintes d'intégrité associées aux données, à l'exception des contraintes de cardinalité, ne sont pas représentables dans le formalisme de base reprenant seulement les types d'entités, d'associations, d'attributs et les connectivités. Elles constituent des annexes au schéma, sous forme d'expressions mathématiques ou logiques.

Les valeurs par défaut : L'expression des valeurs par défaut ou dérivées s'effectue de la même façon, en dehors du dessin des concepts.

Les cas d'exception : La rigidité du modèle E/A empêche également de traiter les cas d'exception de façon naturelle. En effet, certaines occurrences de types d'entités possèdent parfois des propriétés supplémentaires par rapport à la structure de base. Or, le schéma ne contient pas de représentation des occurrences et ne peut donc pas attribuer de propriétés spécifiques à une occurrence donnée. Celles-ci doivent quand-même être reprises comme attributs du type, avec un caractère facultatif (exemple dans un fichier du personnel d'une société : l'attribut nom-de-jeune-fille du type d'entité EMPLOYE).

Une autre possibilité est d'exploiter l'association de sous-typage, à laquelle est associée l'héritage des propriétés. Nous y reviendrons lorsqu'il sera question des inférences.

Le texte et le temps : Il n'existe pas non plus de technique particulière appropriée à la représentation du texte ou du temps. Ces deux notions peuvent être exprimées grâce à certains artifices :

Le texte, dans notre modèle juridique, est un type d'attribut qualifiant les types d'entités JUGEMENT, FAIT, et STANDARD. Il pourrait nécessiter un type d'entité séparé des autres et y être associé par un type d'association.

Un texte pourrait aussi être un ensemble de lignes, étant elles-mêmes des ensembles de mots, éventuellement encore formés de caractères, ou un ensemble d'actions ayant un sujet, un objet et des compléments circonstanciels. Ces notions peuvent être aisément représentées par des entités distinctes reliées par des associations de décomposition ou de qualification.

La conception d'une base de données historique, c'est-à-dire qui conserve les états successifs des données, ne peut se faire que par l'intermédiaire de la notion de temps.

Celle-ci fait l'objet de recherches approfondies [KLOP 81] dont il ne sera pas question ici.

Possibilité de raisonner directement sur le schéma de façon performante : Pour obtenir un système performant travaillant sur le schéma des données, il est nécessaire de le transformer en une structure acceptable par un SGBD, qu'il soit hiérarchique ou relationnel. L'idéal serait de disposer d'un langage orienté-objet, qui manipulerait les entités et les associations comme des objets connus.

La technologie ne permet pas actuellement de raisonner directement sur un schéma de type Entité/Association, c'est-à-dire d'effectuer des inférences sur ses occurrences pour en déduire des résultats. Le modèle ne possède d'ailleurs pas réellement de règles d'inférence autres que la déduction des propriétés associées aux entités par les types d'association ou les types d'attributs. Il est très pauvre en ce qui concerne la détection d'entités ou d'associations à partir de propriétés observées sur des objets du monde réel.

-Le modèle présente la possibilité de définir des relations de sous-typages, lesquelles signifient que les entités sous-types héritent de toutes les propriétés (attributs et associations) du type d'entité dont elles dépendent.

Par exemple, nous pourrions définir le type d'entité FAIT&STANDARD, avec les attributs "n.identifiant", "texte", "polarisation", et ses sous-types FAIT, ayant comme attribut supplémentaire "texte-nië", et STANDARD, ayant comme attribut supplémentaire "état". Ces deux types sont aussi caractérisés, par héritage, par les attributs associés au type d'entité FAIT&STANDARD.

En dehors de cet aspect du modèle, il est impossible d'effectuer automatiquement d'autres déductions sur le schéma.

Les occurrences : Le formalisme ne possède pas non plus de moyen de représenter les occurrences des types.

L'explication en est sans doute la suivante : le schéma Entité/Association est utilisé uniquement au niveau conceptuel pour spécifier les classes d'objets et de relations existant dans le domaine étudié. Ces classes permettent de ranger les informations collectées auprès des experts, de façon à n'oublier aucune donnée reçue et à communiquer plus facilement avec l'utilisateur.

Mais son exploitation dans un système réel nécessite des étapes successives de transformation en une base de données selon des règles déterminées, et une programmation élaborée autour de ces concepts.

Le schéma E/A n'est donc qu'une partie d'un système. Il serait pourtant intéressant de manipuler sous le même formalisme des données et des traitements, d'effectuer des interrogations aussi bien que des comparaisons ou des inférences, et de déduire des sous-structures.

Après chaque présentation d'autres formalismes de représentation des connaissances, nous tenterons de les comparer au modèle Entité/Association, selon les quelques critères énoncés dans ce chapitre.

Cette étude nous permettra par la suite de proposer une nouvelle piste d'orientation pour la représentation des connaissances du modèle juridique et, par conséquent, pour l'implémentation du prototype d'aide à la décision judiciaire.

CHAPITRE 10 : Formalismes de représentation des connaissances.

Section 10.1. : Représentation des connaissances.

Tout au long de ce chapitre, nous analysons les principaux formalismes de représentation des connaissances utilisés dans un domaine particulier de l'Intelligence Artificielle, celui des Systèmes Experts.

Pour rappel, un système expert est "un système permettant la résolution de problèmes dans un domaine spécifique (médecine, géologie, chimie, etc.) en utilisant une base de connaissances acquise auprès d'experts du domaine et un mécanisme de raisonnement caractéristique de la façon de raisonner des experts. (...). Il fournit à la demande du praticien, soit des solutions à ses problèmes, soit l'explication de sa façon de raisonner." [PIES 81]

La création de la base de connaissances propre au domaine exploité nécessite l'implémentation d'une technique de formalisation de ces connaissances qui seront par la suite utilisées par le système expert pour répondre aux questions du praticien.

A. Barr & E. A. Feigenbaum [BARR 82] définissent la représentation des connaissances en IA comme "une combinaison de structures et de procédures qui, utilisées de façon correcte dans un programme, mèneront au comportement «connaissable»".

Dans cette définition, nous pouvons déceler les deux tendances opposées existant parmi les chercheurs de l'IA : les procéduralistes et les déclarativistes.

La représentation procédurale de la connaissance est, par exemple, celle que nous employons habituellement dans nos programmes classiques.

Dans ce type de systèmes, les fragments de connaissances sont organisés de manière explicite et non déterministe, en fonction du raisonnement à suivre, c'est-à-dire dans un ordre fixé par la structure de contrôle. Les procédures sont de petits programmes qui savent comment spécifier les choses et comment procéder dans des situations bien déterminées.

Cette technique facilite le codage et la compréhensibilité du raisonnement.

Par contre, la **représentation déclarative** sépare entièrement les connaissances du mécanisme de déduction, et supprime la notion d'ordre.

Les avantages de cette méthode sont principalement la modularité, la modifiabilité des données et des fonctions et donc une plus grande flexibilité du système.

Il est possible de répartir les différents formalismes connus dans les systèmes experts selon ces deux catégories :

La **représentation procédurale** est celle des automates finis et des programmes. Le calcul des prédicats, les règles de production et les réseaux sémantiques sont des types de représentation déclarative des connaissances.

Une troisième catégorie, que l'on peut qualifier de mixte, puisqu'elle mêle les deux premières, connaît un certain essor. Les formalismes qui s'y rattachent sont principalement les "frames" et le "scripts".

Il est inutile de rappeler les caractéristiques des automates finis, qui sont des protocoles d'actions prises en fonction de l'état courant et de règles de décision permettant de passer de cet état dans un autre.

De même pour les programmes, qui utilisent les techniques de séquencements d'actions, contrôlées par des sauts conditionnels ou inconditionnels, des appels directs de procédures, etc.

Par contre, il serait bon de parcourir plus en détail les autres formalismes cités.

Les sections suivantes seront consacrées à l'analyse du calcul des prédicats, des règles de production, des réseaux sémantiques, des frames et des scripts.

Après chaque définition, nous tenterons d'appliquer la technique aux concepts du modèle du raisonnement et de l'argumentation judiciaire.

Section 10.2. : Le calcul des prédicats.

10.2.1. Définition du calcul des prédicats.

10.2.1.2. Eléments du modèle. Définitions.

Il s'agit ici plus précisément de la logique des prédicats du premier ordre, dont le langage Prolog est une matérialisation.

Ce formalisme répond au besoin de parler d'objets, et de relations entre objets, et de généraliser ces relations à des classes d'objets.

Ses éléments de base sont les prédicats, appliqués à un nombre spécifique d'arguments, les variables, constantes et fonctions, et les quantificateurs universel et existentiel, ajoutés aux opérateurs ($\&$, \vee , \Rightarrow , \Leftarrow , \sim) de la logique propositionnelle. Chaque prédicat appliqué à un argument définit une classe ou un tri d'ensemble.

Pour simplifier l'explication, nous partirons d'un exemple :

L'affirmation vraie

Tous les oiseaux ont des ailes

sera représentée par l'expression :

$\forall x : \text{oiseau}(x) \Rightarrow \text{adesailles}(x)$

où \forall est le quantificateur universel,
x, une variable,
oiseau et adesailles, des prédicats
 \Rightarrow , l'opérateur d'implication.

Le prédicat oiseau, appliqué à la variable x , oiseau(x), signifie que x est un oiseau.

Nous obtenons donc

Pour tout x tel que x est un oiseau, alors x a des ailes.

Cette expression est une **expression bien formée**, c'est-à-dire une connection d'expressions bien formées (ebf) ou de quantificateurs et de littéraux. Les littéraux sont des formules atomiques ou la négation de formules atomiques, qui sont elles-mêmes formées de prédicats et de termes (objets, variables, fonctions) [WINS].

A toute ebf est associée une des deux valeurs de vérité vrai ou faux.

10.2.1.2. Inférences.

Ce formalisme va nous permettre d'effectuer des déductions, et par conséquent, de dériver de nouvelles expressions, grâce à des règles d'inférence.

La plus connue est la règle du **modus ponens** :

$$((P \Rightarrow Q) \ \& \ P) \Rightarrow Q$$

En voici un exemple concret, utilisant également le principe d'unification :

SI $\forall x : \text{oiseau}(x) \Rightarrow \text{adesailes}(x)$

et oiseau(TITI)

ALORS adesailes(TITI)

Par unification, la variable x est instantiée à la constante TITI dans toute l'expression. Le membre de gauche de l'implication étant ainsi vrai, on en déduit, par le modus ponens, que celui de droite l'est également.

10.2.1.3. Critique générale.

La règle du modus ponens est un cas particulier d'une règle plus générale, le principe de résolution de Robinson :

SI $P \vee Q$

et $\neg P \vee R$

ALORS $Q \vee R$

Il a été démontré que ce principe garantissait la correction de la déduction à un point qu'aucun autre formalisme n'a atteint [WINS²⁴].

La logique du premier ordre est également complète, car toute affirmation vraie peut être prouvée à l'intérieur du formalisme.

Ceci constitue évidemment un avantage considérable.

Le calcul des prédicats présente également une grande capacité d'explication de la solution donnée, puisqu'il suffit alors de remonter la chaîne des règles utilisées, en utilisant le principe de résolution dans le sens inverse.

La connaissance sera organisée en deux sous-ensembles : celui des faits, exprimant des affirmations au sujet des constantes du domaine d'expertise, et celui des règles, donnant des relations plus générales entre les prédicats.

L'inconvénient majeur du calcul des prédicats est le risque d'explosion combinatoire dû à un nombre élevé de règles.

Toutes les connaissances ne sont pas exprimables en logique du premier ordre, mais nécessitent une logique d'ordre supérieur.

De plus, la logique classique est indépendante par rapport au contexte, ce qui a mené des chercheurs à étudier d'autres types de logiques, comme la logique modale, par exemple, qui met en évidence les notions de possibilité, nécessité, dépendantes du monde dans lequel on se trouve.

Une analyse plus approfondie de ce formalisme nécessite son application à un cas en dimensions réelles. Notre modèle juridique va servir de base à cette étude.

10.2.2. Application du calcul des prédicats au modèle juridique.

10.2.2.1. Schéma conceptuel.

L'application du formalisme de la logique des prédicats au modèle de l'argumentation judiciaire a été développé lors de la conception du programme réalisé en Micro-Prolog.

Dans ce cas, nous avons mené l'implémentation de la base de faits à partir du schéma Entité/Association (Chapitre 9), en passant par le formalisme relationnel et sa traduction directe en calcul des prédicats.

Une représentation logique simplifiée et indépendante du raisonnement suivi dans les fonctions a également été donnée à la section 7.2., pour formaliser le concept de contexte d'application d'un standard.

Comme pour le modèle du raisonnement jurisprudentiel, il s'agit de logique propositionnelle, et non du premier ordre.

L'adaptation du modèle juridique au calcul des prédicats pose un problème très important de formation de l'utilisateur juriste qui désire introduire, par exemple, de nouvelles compositions d'un jugement dans la base de données. Jusqu'à présent, il utilise, pour les compositions, la logique des propositions, ce qui demande déjà un certain effort de compréhension de la logique, et le langage naturel, pour enregistrer les textes des faits ou des jugements à l'état brut.

Il devrait, dans le cas d'une formalisation sous forme de calcul des prédicats, transformer lui-même ces textes en expressions bien formées avant de les enregistrer. A moins, bien sûr, que le système de possède lui-même un traducteur de langage naturel, ce qui semble encore inespéré dans l'état actuel des recherches.

Néanmoins, pour donner une idée de ce que pourrait être la formalisation d'une composition en logique des prédicats, voici quelques exemples de formalisation de faits repris dans la base actuelle des jugements :

Soient C appartenant à l'ensemble S(ociétés) : le Créancier,
DP " " " " S : le Débiteur Principal,
CO " " " " P(ersonnes) : la Caution

originaire,

$\text{credit}(x, y) : x \in S$ offre un crédit à $y \in S$,
 $\text{caution}(x, y, z) : x \in P$ donne sa caution à $y \in S$ pour le
crédit de $z \in P$,
 $\text{modif}(x) : \text{il y a une modification de } x \in S$,
 $\text{develop}(x) : x \in S$ se développe,

Fait n-9 : Le créancier a l'habitude de solliciter le renouvellement de la caution après un changement de forme de la société débiteur principal.

$\forall x \in S, \exists y, z \in P :$

$(y <> z) \ \& \ ((\text{credit}(C, x) \ \& \ \text{caution}(y, C, x)) \Rightarrow (\text{modif}(x) \Rightarrow \text{caution}(z, C, x)))$

Fait n-10 : Il y a eu un développement des activités de la société débiteur principal après sa transformation.

$\text{modif}(DP) \ \& \ \text{develop}(DP)$

Fait n-12 : Le cautionnement était nécessaire pour l'obtention du crédit.

$\text{credit}(C, DP) \Rightarrow (\exists x \in P : \text{caution}(x, C, DP))$

10.2.2.2. Conclusion.

En conclusion, nous pouvons affirmer que la logique paraît une manière naturelle d'expression de certaines notions, qu'elle est précise, sa signification étant déterminée par des méthodes standards, qu'elle est flexible, un fait pouvant être représenté peu importe son usage, qu'elle est modulaire, et régie par des règles garantissant sa correction et sa complétude.

Mais elle ne procure pas de facilités adéquates pour définir des structures plus complexes et les experts du domaine connaissent des difficultés pour utiliser le calcul des prédicats ou pour comprendre la connaissance qui y est exprimée.

- Par conséquent la transformation en logique des prédicats du premier ordre nécessite un travail en profondeur avec les juristes, qui ne nous a pas paru utile, ni réalisable dans le cadre de ce mémoire.

10.2.3. Comparaison du calcul des prédicats avec le modèle Entité/Association.

Reprenons un à un les critères de comparaison relevés dans le chapitre 10 pour analyser la logique des prédicats.

Simplicité du formalisme : La logique des prédicats est un mode de représentation qui nécessite malgré tout un assez grand nombre d'éléments : variables, prédicats, constantes, fonctions, quantificateurs, etc. Ces composants doivent être agencés de manière précise pour former des EBF. Nous ne pouvons donc pas prétendre que ce formalisme est réellement simple et facile à utiliser et à apprendre.

Visualisation et communication avec l'utilisateur : Si nous reprenons l'exemple des compositions d'un jugement, ou des contextes d'application d'un standard, nous pouvons constater que leur formalisation logique semble tout à fait appropriée à leur nature. En effet, la logique leur donne la visualisation la plus évidente.

Par contre, elle ne permet pas de représenter de façon claire les informations fixes relatives aux faits, standards ou jugements, telles la polarisation ou la date.

De plus, comme nous l'avons signalé à la section 10.2., les expressions logiques se multiplient très rapidement lorsque les connaissances augmentent. L'aspect communicatif du calcul des prédicats en est donc fortement compromis.

Capacité de structuration des informations : l'accès aux propriétés attribuées aux concepts, ou à leurs associations avec d'autres dépend de la façon dont sont écrites les expressions logiques. Mais il est plus que probable qu'il y aura une certaine dispersion des données et que leur accès nécessitera, l'utilisation de plusieurs règles situées en des endroits différents, avec le risque d'oubli qui en découle.

Expression de relations n-aires : Les relations n-aires ne posent aucun problème en logique du premier ordre, puisqu'il suffit de les représenter sous forme de prédicats n-aires, dont les arguments sont les concepts associés par la relation.

Extensibilité : Le calcul des prédicats débouche sur une représentation du réel facilement extensible. L'ajout ou le retrait de certaines expressions ne devrait en aucun cas modifier le sens des informations déjà codées.

Les traitements : La preuve a été faite de l'adaptation de la logique à la programmation : Prolog est un outil dont la syntaxe, et les règles de raisonnement sont celles de la logique, puisqu'il travaille avec des clauses de Horn, cas particulier des EBF, et le principe de résolution de Robinson.

Mais, pour réaliser réellement des traitements d'information, on a ajouté à la logique, des prédicats spéciaux tels les impressions, ajouts, suppressions de règles, etc. La logique améliorée peut donc devenir aussi un langage de programmation efficace.

Les contraintes d'intégrité : Les contraintes d'intégrité d'un modèle conceptuel trouvent leur meilleure expression dans la logique, car ce sont des affirmations générales portant sur tous les éléments d'un ensemble, aisément traduisibles en EBF portant sur des variables.

Les valeurs par défaut : Il est possible d'exprimer en logique que, pour tous les éléments tels qu'aucune valeur n'existe encore, c'est une valeur donnée qui prévaut.

Les cas d'exception : L'utilisation du quantificateur existentiel permet de définir des règles valables pour certains éléments d'un ensemble, mais pas pour tous, ce qui correspond bien à des cas d'exception.

Le texte : Le texte à l'état brut paraît difficilement représentable dans la logique des prédicats, sauf éventuellement comme argument d'un prédicat.

La traduction d'un texte en logique des prédicats pose de gros problèmes de sémantique auquel s'attaquent toujours aujourd'hui les logiciens. Y a-t-il une représentation logique correspondant à chaque expression du langage naturel ? Les seize opérateurs logiques existants sont-ils suffisants ? La réponse à ces questions n'a pas encore réellement été donnée.

Le temps : Cette notion est absente du formalisme de la logique classique. Il existe des tentatives de définition d'une logique temporelle, c'est-à-dire une logique dans laquelle le temps est un concept particulier, et donc traité différemment.

Les occurrences : Le calcul des prédicats manipule des éléments appelés constantes, qui représentent des occurrences d'ensembles. C'est à l'aide de ces éléments qu'est établie la base des faits d'un système de programmation logique. Dans une même règle peuvent intervenir à la fois des constantes et des variables, signifiant alors que l'expression est vraie pour tous les éléments d'un ensemble. Il est donc capable de traiter les cas particuliers comme les cas généraux.

Possibilité de raisonner directement sur le schéma : Le principe de résolution constitue une base sûre pour l'obtention de résultats à partir d'un ensemble d'expressions logiques. Mais l'implémentation qui en a été faite, nous pensons notamment à Prolog, restreint le raisonnement aux clauses de Horn, ce qui mène parfois à des erreurs sémantiques [KOWA].

Ceci mis à part, comme il a été dit plus haut, la résolution garantit une solution correcte à partir des connaissances décrites. La plupart du temps, les systèmes proposent des performances appréciables au niveau du temps de réponse également.

Section 10.3. : Les règles de production.

10.3.1. Définition des règles de production.

10.3.1.1. Définitions.

Le schéma général d'une règle de production est le suivant :

SI <condition> ALORS <action>

signifiant que

"Si cette condition existe, alors cette action est appropriée".

Un système de production travaille en relation avec un buffer, ou mémoire à court terme, appelé "espace de travail" ou "contexte".

10.3.1.2. Inférences.

Si la condition existe dans le contexte, l'action peut être exécutée.

Cette action est en général elle-même une modification de la mémoire de travail, mais elle peut aussi être une question à l'utilisateur, une procédure, ou une interaction avec un terminal.

Le système progresse vers la solution en "matchant" (appliquant) un côté de la règle (gauche, si chaînage avant, droit si chaînage arrière) au contenu du contexte. Le plus spécifique aux règles de production est le chaînage avant, le chaînage arrière étant le plus souvent réservé au calcul des prédicats.

Lorsque plusieurs règles répondent à cette condition, une seule est choisie en fonction d'une stratégie déterminée [McDe~~W~~].

10.3.1.3. Critique générale.

Les avantages de cette technique sont sa modularité, son caractère naturel et son uniformité de structure.

Les règles de production peuvent être facilement comprises par les experts et ont un pouvoir d'expression suffisant pour représenter un ensemble utile de règles d'inférence dépendantes du domaine et des spécifications de comportement.

De plus, les conditions d'application sont explicites et les interactions entre les règles sont minimisées.

Comme les règles du calcul des prédicats, les règles de production sont une base aisée pour l'explication de la solution obtenue, dans des termes compréhensibles par l'utilisateur, puisqu'elles respectent sa logique.

L'expert familier avec le système peut être capable de formuler lui-même de nouvelles règles si nécessaire.

Ces caractéristiques impliquent une grande facilité de communication et de modification de la base de règles.

Elles sont également tout à fait indiquées pour l'enregistrement d'un "savoir-faire" ou d'une connaissance impérative.

Les adeptes du parallélisme trouveront aussi dans ce formalisme l'occasion de réaliser des performances appréciables. [HAYE 85]

Le revers de la médaille est constitué par un manque d'efficacité lors d'un grand nombre de règles, et par une certaine opacité due à l'absence de représentation de connaissance algorithmique.

Les règles de production ne possèdent pas non plus de fondement théorique permettant de dire si un problème est soluble, si un ensemble de règles est consistant et complet, ni de méthode d'organisation réalisant d'office ces deux derniers objectifs.

Puisque les règles ne sont pas appelées explicitement par d'autres règles, nous ne sommes jamais sûrs des effets de bord qui peuvent survenir de l'addition ou de la modification d'une règle. Certaines connaissances ne peuvent pas facilement être représentées sous forme de règles de production.

La sous-section suivante tente d'appliquer ce formalisme au modèle étudié.

10.3.2. Application des règles de production au modèle juridique.

10.3.2.1. Schéma conceptuel.

Le contexte d'application d'un standard est une notion assez facile à représenter sous forme de règles de production, puisque celles-ci ne sont pas très éloignées du formalisme de la logique propositionnelle.

Dans le but de trouver la meilleure argumentation en fonction de faits et/ou d'une polarisation introduits, les règles nécessaires pourraient être les suivantes :

IF argumentation & (F1 & F2 & F3) & polarisation +
THEN CN1.

IF argumentation & (-F4 & F5) & polarisation +
THEN CN2.

IF argumentation & (S1 & F6) & polarisation +
THEN CS1.

IF argumentation & (F7 & -F8) & polarisation +
THEN CS2.

IF argumentation & CN1 & CS1
THEN S2 dans J1.

IF argumentation & CN1 & CS2
THEN S2 dans J2.

IF argumentation & CN1
THEN S2.

IF argumentation & CN2
THEN S1.

F1, F2, F3, F4, F5, F6, F7 et F8 sont des faits, S1 et S2 des standards, CN1 et CN2 des contextes nécessaires, CS1 et CS2 des contextes spécifiques, et J1 et J2 des jugements.

Les faits et la polarisation se trouvant dans l'espace de travail seront "matchés" avec les conditions des règles. Si plusieurs règles vérifient cette condition, nous choisirons la première, qui nous donnera un contexte, puis la suivante, etc. Ces contextes sont conditions d'application des standards dans les règles suivantes. Ils seront enregistrés dans l'espace de

travail, dans l'ordre défini pour les règles. Les réponses finales seront les premiers éléments de données contenus dans l'espace de travail.

Des règles supplémentaires devraient être ajoutées pour déterminer les faits à prouver pour appliquer un standard donné :

IF consultation & S2

THEN CN1 OR (CN1 & CS1) OR (CN1 & CS2).

IF consultation & CN1

THEN F1 d'origine loi & F2 d'origine loi & F3 d'origine loi.

IF consultation & CN2

THEN -F4 d'origine doctrine & F5 d'origine cour de cassation.

IF consultation & CS1

THEN S1 & F6 dans J1.

IF consultation & CS2

THEN F7 & -F8 dans J2.

Ce sont donc des occurrences de contextes d'application d'un standard qui sont représentées, de manière à pouvoir être utilisées immédiatement, dans un but déterminé.

L'indication de la fonction pour laquelle on interroge la base est nécessaire dans la condition de chaque règle, car plusieurs règles peuvent travailler sur les mêmes concepts, sans donner les mêmes renseignements.

Les informations générales au sujet des notions de fait, standard, jugement, contextes nécessaire et spécifique, peuvent être formalisées selon des règles du même type que celle de l'exemple suivant :

IF S1 & infos

THEN n·identifiant, polarisation +, <texte>, état affirmé.

On reprendra les compositions d'un jugement sous deux formes, selon que l'on désire connaître le contenu des compositions d'un jugement ou les jugements comprenant certaines compositions :

IF J1 & contenu

THEN C1 & C2 & C3.

IF C1 & contenu

THEN ((F1 & F2) => S1).

IF C2 & contenu

THEN ((S1 & F3 & F4) => S2).

IF C3 & contenu

THEN (F5 => S3).

et

IF C1 & jugement

THEN J1 & J2.

IF C2 & jugement

THEN J1.

IF C3 & jugement

THEN J2.

L'ensemble de règles de production ci-dessus ne prétend pas représenter complètement l'information contenue dans le schéma -E/A, car il faudrait encore traduire les relations de personnes et de classes, et pouvoir interroger les jugements sur base d'éléments de faits ou de standards.

De plus, les associations et les relations d'attributs d'un schéma E/A pouvant être parcourues dans les deux sens, il est nécessaire d'écrire chaque fois deux règles pour ne perdre aucune information.

10.3.2.2. Conclusion.

Le système de production donné ici ne constitue donc qu'un projet, qu'une direction vers laquelle s'orienter si l'on décidait de choisir les règles de production comme technique de représentation des connaissances dans le SIAD judiciaire.

Mais, comme cette tentative de formalisation le démontre déjà, dès que le système concernera un grand nombre de jugements et de standards, le risque d'explosion combinatoire deviendra insurmontable.

Un second inconvénient déjà soulevé est la dépendance des règles par rapport au contexte, à la fonction susceptible de les utiliser. Dans le cas des compositions, par exemple, si l'on veut connaître le contenu d'une composition, il faut ajouter l'antécédent "contenu" dans la condition, tandis que lorsqu'on désire les jugements dans lesquels elle intervient, c'est le mot "jugement" qu'il est nécessaire d'ajouter. Il faut remarquer que cette critique est vraisemblablement valable également pour le calcul des prédicats, bien que nous n'ayons pas approfondi son application au modèle juridique.

Nous pouvons dès lors imaginer que lors de la création de nouveaux traitements, il faudrait ajouter de nouvelles règles concernant peut-être des concepts déjà utilisés ailleurs, mais dans lesquelles on modifie le signal du contexte.

De même, si l'on ajoute une occurrence d'une entité quelconque, il est indispensable d'ajouter au moins une règle la concernant dans chaque sous-ensemble de règles relatif à une fonction utilisant cette notion.

En plus de cette dépendance par rapport au contexte, une autre dépendance existe vis à vis de la stratégie de choix des règles à appliquer, qui peut influencer l'ordre et même le contenu des règles. Nous pouvons en conclure que les règles de production ne sont pas un formalisme de représentation purement déclaratif !

En effet, que l'on choisisse la première ou la dernière règle, celle qui a le plus d'antécédents ou celle qui en a le moins, le système de production doit présenter un visage différent.

Il n'existe donc pas réellement une représentation du modèle juridique selon le formalisme des règles de production, qui soit totalement indépendante du mécanisme de raisonnement requis par le système.

Ces derniers points constituent des arguments très importants en défaveur du formalisme des règles de production.

Les règles de production ne semblent pas non plus adéquates pour définir des termes et décrire des objets et les relations statiques entre ces objets.

Malgré les avantages soulevés dans la sous-section précédente, nous sommes forcés de remarquer encore une fois que la pratique d'un problème réel rend ce formalisme bien complexe à utiliser.

10.3.3. Comparaison des règles de production avec le modèle Entité/Association.

Simplicité du formalisme : Seule la forme de l'implication SI...ALORS... est à retenir lorsqu'on désire exprimer des connaissances sous forme de règles de production. C'est donc une structure très simple.

Visualisation et communication avec l'utilisateur : Ce critère vient d'être abordé à la section 11.3.2. : le caractère naturel de la représentation est limité par le risque d'explosion combinatoire résultant d'un grand nombre de connaissances.

Capacité de structuration des informations : L'analyse faite à propos du calcul des prédicats reste valable pour les règles de production, avec toutefois l'inconvénient supplémentaire dû à la contrainte de formalisation des règles sous forme d'implication, qui rend l'accès aux données parfois assez artificiel.

Expression de relations n-aires : Le formalisme des règles de production n'est pas particulièrement adapté au codage des connaissances relationnelles, bien qu'il ne soit pas impossible de les exprimer à l'intérieur d'un des membres d'une règle. Tout dépend de la syntaxe acceptée par le système dans lequel s'effectue l'implémentation. Les associations n-aires peuvent aussi être transformées en relations binaires grâce à certains artifices de représentation.

Extensibilité : Nous avons déjà fait remarquer que l'ajout de règles de production peut se faire aisément par ajout de modules relativement distincts, mais qu'à l'intérieur d'un module, des effets de bord dus à l'ordre des règles, par exemple, risquent de survenir, si l'on n'y prête pas suffisamment attention. Ce risque se retrouve dans les implémentations qui sont faites du calcul des prédicats (Exemple : Prolog), puisque l'ordre d'écriture des règles, par exemple, peut y avoir de l'importance.

Les traitements : L'action contenue dans le second membre d'une règle de production est une modification de la mémoire de travail, mais aussi de la base de données, une procédure ou une interaction. Des procédures de traitement peuvent donc être parfaitement intégrées à la représentation du modèle des données.

Les contraintes d'intégrité et les valeurs par défaut : A première vue, les règles de production ne paraissent pas adaptées à la formalisation des contraintes d'intégrité ou des valeurs par défaut. En effet, le résultat d'une règle est toujours une action sur la base ou au terminal, et non une affirmation générale relative à une collection de données. Une étude plus approfondie des règles de production permettrait peut-être de donner une solution à ce problème.

Les cas d'exception : Les règles de production sont faites pour la représentation de connaissances concernant un savoir-faire ou impératives. Les cas d'exception seront traités de la même manière : SI nous nous trouvons dans telle situation particulière (l'exception), ALORS c'est cette action qui est appropriée.

Le texte : L'implémentation sous-jacente aux règles de production détermine la possibilité de faire intervenir du texte en langage naturel à l'intérieur d'une règle. Ceci ne nous paraît pas irréalisable a priori.

Le temps : Les règles de production ne disposent pas actuellement de technique spécifique de traitement de la notion de temps.

Les occurrences : Un système de production étant un outil de programmation destiné à l'implémentation des fonctions d'interrogation, les règles de production sont d'abord un formalisme de représentation des occurrences, mais on peut imaginer que les concepts puissent être formalisés de la même façon.

Possibilité de raisonner directement sur le schéma : Comme nous venons de le remarquer, les règles de production sont d'abord un langage de programmation, plutôt déclaratif. Elles sont donc écrites de façon à être utilisées directement pour obtenir les

réponses à certaines questions. Il existe tout un mécanisme de raisonnement induit par le formalisme, basé sur une forme du modus ponens, et grâce auquel il est possible d'inférer des résultats automatiquement à partir des règles.

Section 10.4. : Les réseaux sémantiques.

10.4.1. Définition des réseaux sémantiques.

10.4.1.1. Eléments du modèle. Définitions.

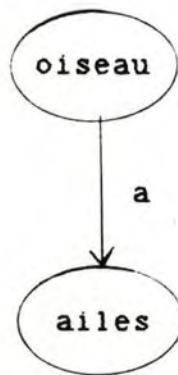
L'objet des réseaux sémantiques est de décrire les éléments d'un univers et leurs interrelations.

Les deux composants du modèle sont les "noeuds" signifiant la présence d'objets, de concepts, de situations ou d'événements, et les "arcs" (orientés) exprimant leurs interrelations.

Comme pour la définition du calcul des prédicats, nous procéderons par des exemples [BAR 82] :

Exemple : **Tout oiseau a des ailes**

sera représenté par :



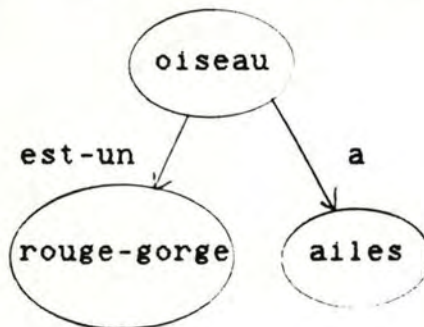
où oiseau et ailes sont des noeuds et a un arc.

Un premier avantage de ce formalisme est la possibilité d'inférer directement les faits relatifs à un objet ou concept à partir des noeuds auxquels il est directement lié, sans recherche dans une grande base de données.

10.4.1.2. Inférences.

La particularité d'un réseau sémantique est d'autoriser des inférences, grâce à la relation spéciale exprimée par l'arc **est-un**, signifiant qu'un concept forme un sous-ensemble d'un autre.

Exemple : Tous les rouges-gorges sont des oiseaux.



De ce schéma, nous pouvons inférer que

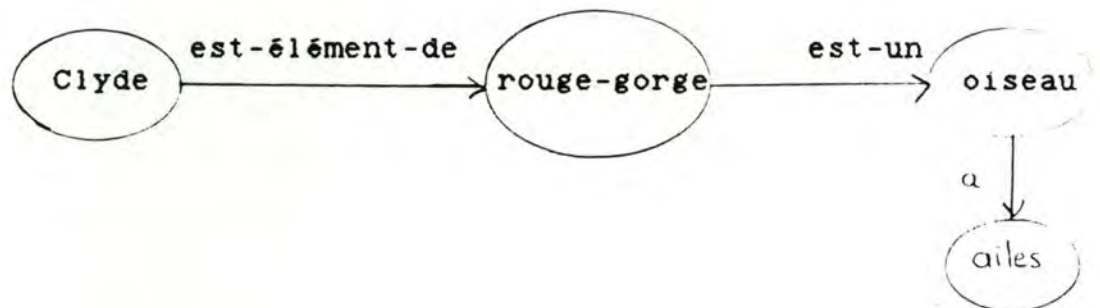
Tous les rouges-gorges ont des ailes

par la hiérarchie d'ensembles liée à l'arc **est-un**, appelée aussi **hiérarchie d'héritage de propriétés**.

En d'autres termes, le noeud d'où part l'arc **est-un** hérite de toutes les propriétés du noeud auquel cet arc aboutit.

Un autre type de relation permet également l'héritage de propriétés : c'est la relation **est-élément-de**.

Par exemple, **Clyde est un rouge-gorge** :



Clyde est un élément de l'ensemble des rouges-gorges, lui-même sous-ensemble des oiseaux. Nous pouvons en déduire que

Clyde est un oiseau

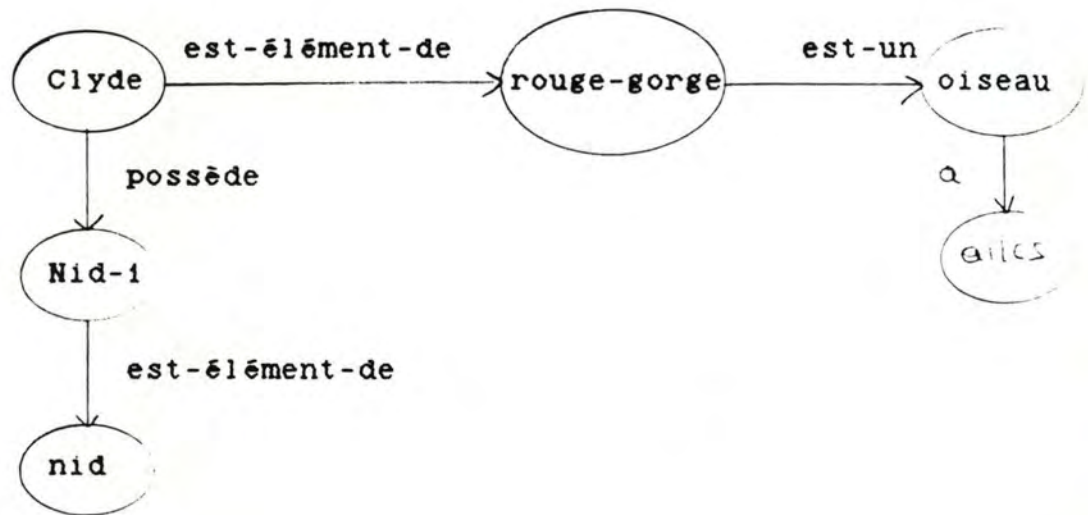
et même que **Clyde a des ailes**,

puisque'il hérite de toutes les propriétés de l'ensemble dont il est élément.

Si l'on veut représenter le fait que

Clyde possède un nid,

il suffit d'ajouter au réseau un arc **possède de** Clyde vers un élément de l'ensemble des nids, **Nid-1**, qui est le nom représentant le nid de Clyde :



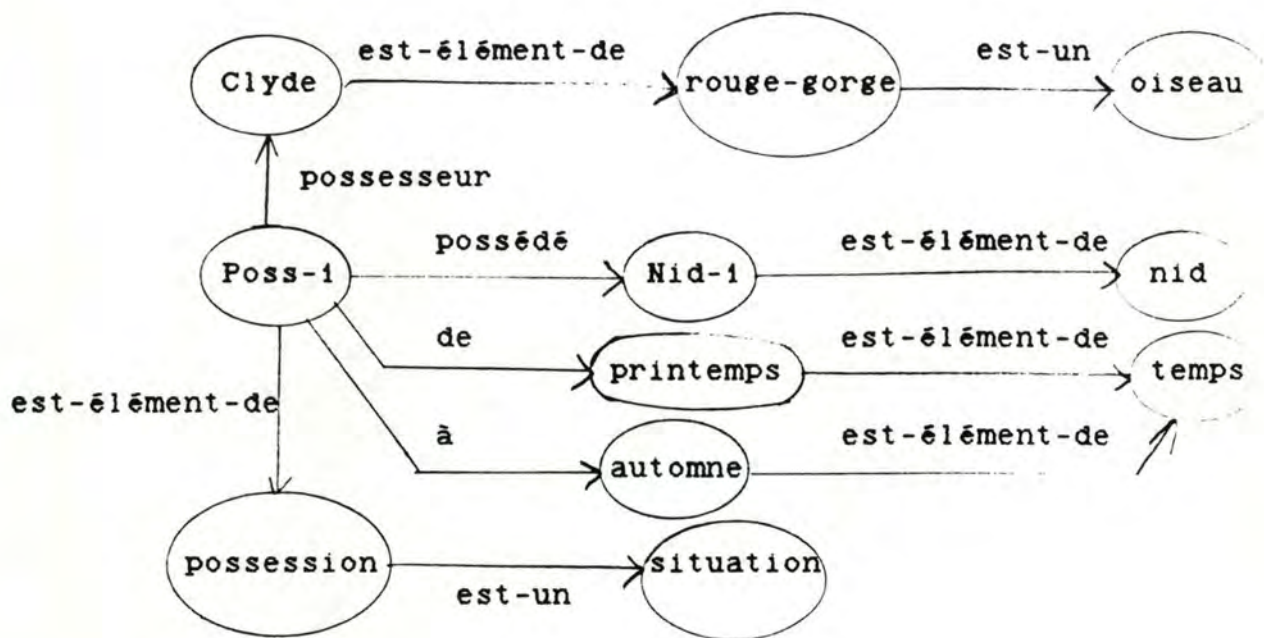
10.4.1.3. Relations plus que binaires (n-aires).

Un des problèmes posés par les réseaux sémantiques est le codage de relations n-aires.

Exemple : **Clyde possède un nid du printemps à l'automne.**

Une solution proposée par [BAR 82] est d'utiliser des noeuds représentant des situations et des actions, et d'en faire partir des arcs spécifiant les arguments de la situation.

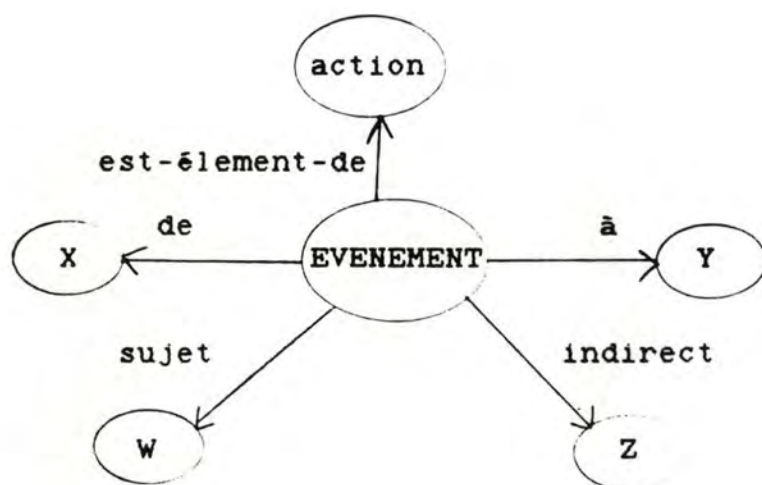
Dans ce cas :



Ce réseau signifie que la notion de **possession** est un sous-ensemble de l'ensemble des **situations**, que **Poss-1** est une occurrence de **possession**, dont le **possesseur** est **Clyde**, le **possédé** **Nid-1**, le moment de départ **printemps**, et celui de fin **automne**, qui sont tous deux des éléments du concept de **temps**.

Nous voyons donc comment des noeuds peuvent spécifier des situations et même des actions.

Ce schéma paraît idéal pour représenter, par exemple, une grammaire. En effet, comme dans une phrase, chaque action est un noeud associé à d'autres noeuds par un arc **sujet**, un arc **objet**, un arc **complément indirect**, en arc **complément de lieu**, de **temps**, etc.



10.4.1.4. Remarques.

Il est utile de remarquer que l'interprétation des réseaux, c'est-à-dire la signification que l'on attribue aux noeuds et aux arcs, ne dépend que du programme qui les manipule. Il n'y a pas réellement de convention à ce sujet.

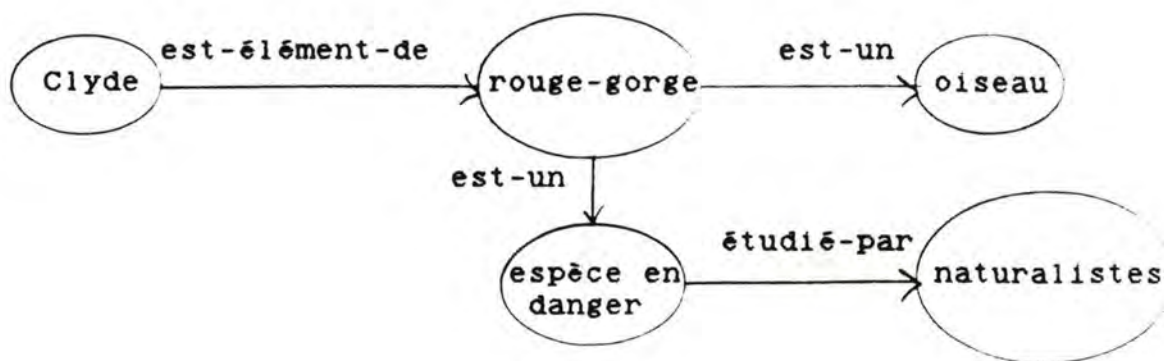
La validité des inférences faites à partir d'un réseau sémantique n'est pas non plus démontrée au même titre que pour la logique. Il reste donc toujours une marge d'erreur à attribuer à chaque déduction.

Essayons maintenant de traduire les phrases :

Le rouge-gorge est une espèce en danger.

Les espèces en danger sont étudiées par les naturalistes.

Nous obtenons le modèle suivant :



Si nous appliquons le principe des inférences sur base des arcs **est-un** et **est-élément-de**, nous pouvons déduire :

Les rouges-gorges sont étudiés par les naturalistes
et Clyde est étudié par les naturalistes.

Or, cette dernière affirmation n'est pas nécessairement vérifiée!

Il faut donc se méfier de la hiérarchie de propriétés portant sur l'arc **est-élément-de**. Certaines propriétés (ex.: a des ailes) peuvent être héritées par tous les éléments de la classe, mais pas toutes (ex.: est étudiée).

Cette remarque pose une limite importante à la théorie des réseaux sémantiques, puisqu'elle remet en question un de ses principes fondamentaux.

Des recherches récentes explorent les différentes façons de rendre la sémantique des structures de réseaux plus précise et de spécifier plusieurs stratégies d'héritage de propriétés (Woods, Hendrix, Brachman, Stefik).

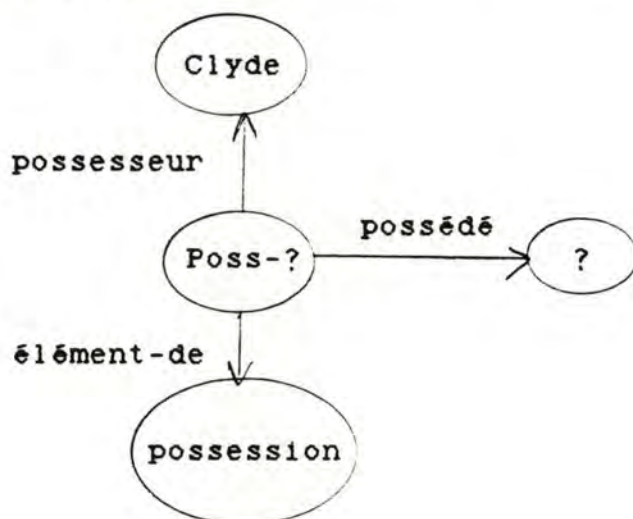
10.4.1.5. Raisonnement.

Comment raisonner sur un réseau sémantique ?

Dans la plupart des systèmes, c'est en utilisant le principe du "matching" que l'on parvient à déduire les résultats espérés.

Voici en quelques mots comment on procède : Une question posée par l'utilisateur est formalisée selon une structure correspondant à un fragment du réseau. Cette structure est "matchée", c'est-à-dire presque superposée, à la base de données réseau, grâce à laquelle les valeurs inconnues de la structure seront déduites.

Dans l'exemple suivant portant sur le dernier exemple de réseau concernant Clyde, la question : **Que possède Clyde ?** est codée par la structure :



représentant une occurrence de possession dans laquelle Clyde est le possesseur.

Après matching sur la base, les ? sont remplacés respectivement par 1, et Nid-1 qui est un élément de l'ensemble des nids.

La réponse à la question est donc **Clyde possède un nid.**

Il existe des stratégies permettant au système de gagner du temps en le conseillant sur le choix du premier élément à matcher (ex: SNIFFER).

10.4.1.6. Critique générale.

- Malgré ces problèmes sémantiques et techniques toujours à l'étude, les réseaux sémantiques constituent une représentation aisée des connaissances relationnelles. Ils sont tout naturellement indiqués dans un domaine de raisonnement basé sur

une taxonomie complexe, et peuvent servir d'aide à la preuve, en mémorisant un problème de façon globale et efficace, et en guidant l'introduction d'éléments nouveaux.

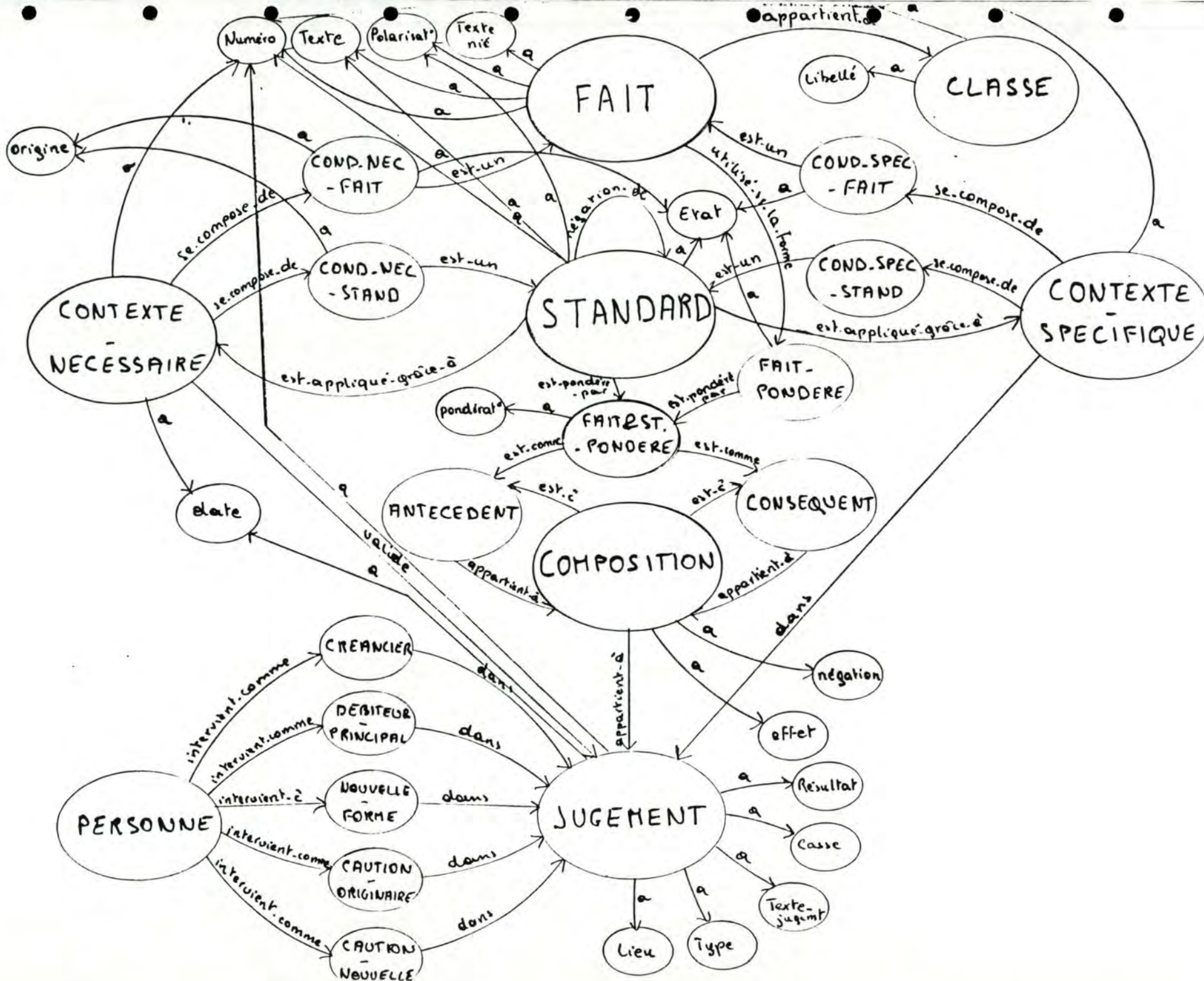
Cependant, ce formalisme n'est pas approprié à tous les types de connaissances, car toutes ne peuvent pas s'exprimer sous forme de réseaux. L'expérience démontre que ces systèmes nécessitent beaucoup de temps machine, et qu'ils éprouvent souvent des difficultés à fournir des explications, ce que tout système expert devrait pouvoir faire.

Les réseaux sémantiques restent pourtant une technique de formalisation très populaire.

10.4.2. Application des réseaux sémantiques au modèle juridique.

10.4.2.1. Schéma conceptuel.

Nous avons traduit le modèle conceptuel global du SIAD juridique, tel qu'il est donné au chapitre 9, sous forme d'un réseau sémantique. Celui-ci reprend les entités et les associations binaires avec attributs sous forme de noeuds, tandis que les associations binaires sans attributs sont représentées par des arcs. Les attributs sont eux-mêmes des noeuds reliés aux entités ou associations qu'ils qualifient.



Nous avons donc choisi dans ce cas de traduire de façon quasi-automatique le schéma Entité/Association en un schéma réseau, puisque le premier est lui-même une formalisation de concepts et de relations.

Le problème qui se pose immédiatement est celui de la traduction d'associations entre deux entités de même type. En l'occurrence, l'association NEGATION-STANDARD reliant deux entités de type STANDARD devrait, si l'on respecte les principes de traduction choisis, être représentée par une boucle. La théorie des réseaux sémantiques accepte-t-elle cette forme de représentation ? Nous n'avons nulle part trouvé de confirmation ou de contre-indication à ce sujet.

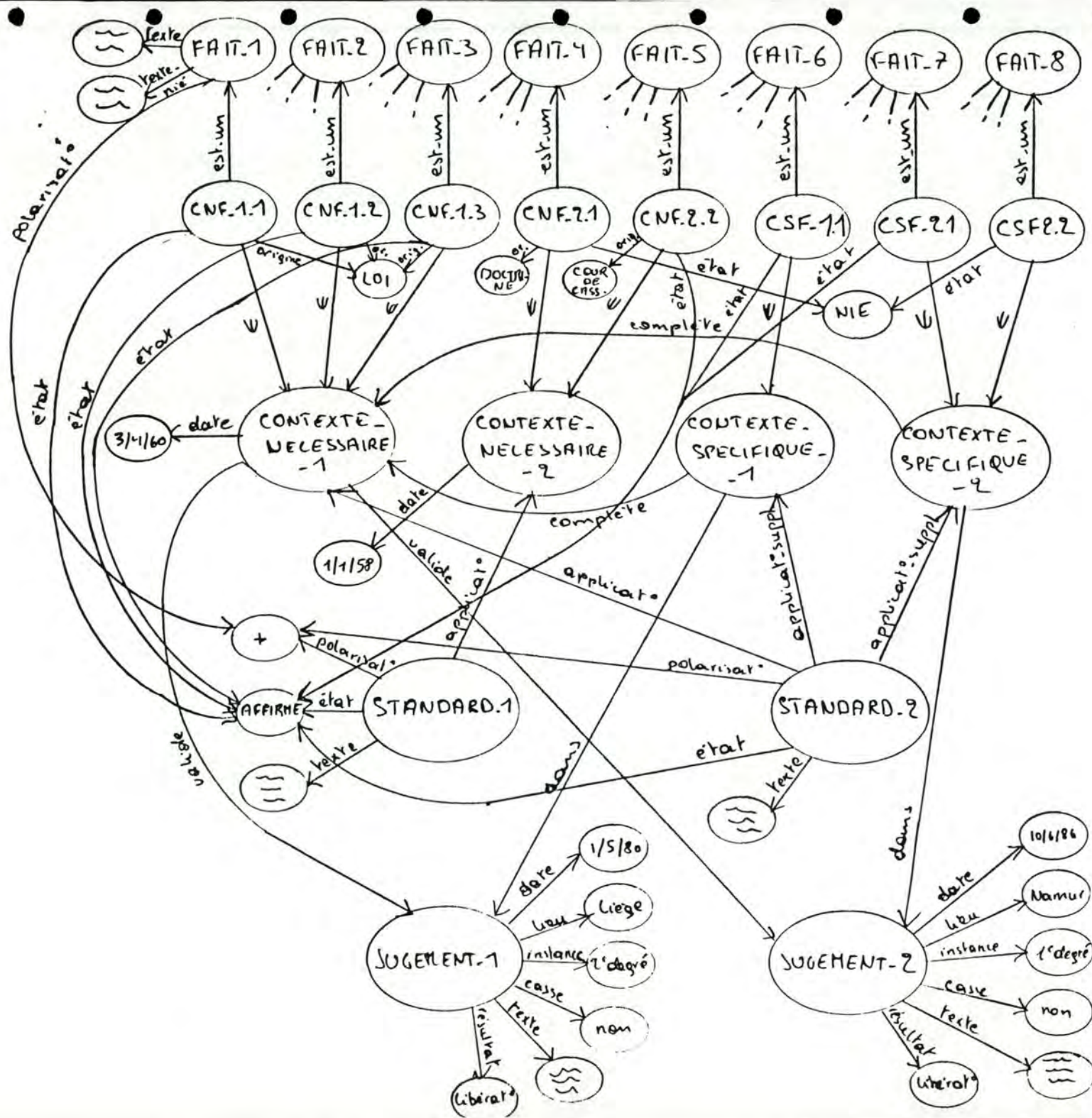
Pour éviter la boucle, un concept "bidon" de négation, associé de deux façons au noeud **standard** par les arcs **est-nié-par** et **nie** peut être ajouté. Malgré tout, ce n'est pas une écriture très naturelle.

Le schéma Entité/Association possède un outil capable, par exemple, de rendre facultatif l'association entre deux occurrences d'entités : les connectivités. Comment exprimer dans le cas d'un réseau sémantique le fait qu'un **standard** affirmé n'a pas toujours de correspondant nié, ou qu'un contexte n'a pas toujours un **standard** comme condition d'application ?

Ces contraintes d'intégrité, comme les autres, doivent être exprimées sous une forme différente (par exemple, sous forme de prédicats), si l'on ne veut perdre aucune information.

Un réseau sémantique, comme nous l'avons vu, permet aussi et surtout la spécification d'éléments d'ensembles. C'est pourquoi nous avons tenté de représenter un exemple de contexte d'application de **standard**, dont chaque noeud est une instance qui pourrait être reliée par un lien de hiérarchie au noeud conceptuel correspondant dans le schéma réseau précédent.

Voici le réseau que nous obtenons pour représenter le contexte d'application déjà traité en 11.3.2 selon le formalisme des règles de production :



10.4.2.2. Conclusion.

Comme chacun peut s'en rendre compte, ce schéma est très peu clair. Imaginons-le maintenant avec deux contextes, et associations-le au réseau des entités ! Cela devient rapidement illisible.

Par conséquent, nous sommes forcés de constater que ce formalisme, si naturel à première vue, devient irréalisable lorsque les notions à représenter deviennent fortement détaillées et interconnectées.

L'apparition de textes en langage naturel constitue encore une difficulté supplémentaire. Il est impossible d'écrire un texte dans un noeud, surtout lorsqu'il devient un peu long.

En analysant un peu plus le dernier réseau, nous pouvons encore nous rendre compte d'une chose :

Comme pour les règles de production, les connaissances introduites dans une base de données sous forme d'un réseau sémantique peuvent être organisées de façon tout à fait anarchique, sans aucun respect d'une structure de base, autre que celle des noeuds et des arcs.

En effet, nous pourrions par exemple ajouter n'importe quelle propriété aux éléments du contexte d'application d'un standard. La configuration en réseau ne la refuserait pas, à moins, bien entendu, que le programme gérant le réseau ne vérifie lui-même, pour chaque noeud, les propriétés qu'il doit ou ne doit pas posséder, en regard du schéma conceptuel du réseau.

Cet inconvénient peut devenir un avantage dans le cas où l'on désire attribuer à un objet une information qu'il possède contrairement aux autres éléments de l'ensemble dont il fait partie, autrement dit dans les cas d'exception.

Toutes les remarques qui peuvent être faites à l'encontre du formalisme des réseaux sémantiques sont certainement dues au fait qu'un réseau est conçu principalement dans le but d'exprimer des occurrences. Or, ce qui nous intéresse également, c'est de pouvoir leur donner une structure, et donc de représenter les concepts sous-jacents aux données. Mais, comme nous l'avons vu dans le cas traité, le schéma des concepts n'induit pas le schéma

des occurrences, c'est-à-dire qu'il ne lui donne pas un cadre organisationnel obligatoire, contrairement au modèle Entité/Association, qui, lui, privilégie les concepts.

En conclusion, nous pouvons affirmer que si le formalisme des réseaux sémantiques est sans doute efficace dans des domaines où les connaissances sont rassemblées en un petit nombre de concepts et fortement hiérarchisées, il ne convient certainement pas idéalement à notre problème juridique, ne fût-ce que par le manque de clarté du schéma auquel il donne lieu.

10.4.3. Comparaison des réseaux sémantiques avec le modèle Entité/Association.

Le formalisme des réseaux sémantiques présente certains avantages et inconvénients par rapport au schéma Entité/Association. Nous allons les mettre en évidence selon les critères habituels :

Simplicité du formalisme : Les noeuds et les arcs sont les seuls composants du modèle. Leur signification est clairement définie. Nous pouvons par conséquent affirmer que les réseaux sémantiques sont un formalisme simple.

Visualisation et communication avec l'utilisateur : Dans un premier temps, nous aurions été tentés de dire que la formule du réseau sémantique présentait les faits sous une forme claire et précise, puisque la seule interprétation impliquait la notion de relations entre concepts, ce qui paraît une façon naturelle de représenter la réalité.

Pourtant, lorsque le nombre de concepts devient aussi élevé que celui du modèle juridique, par exemple, nous sommes forcés de constater que, les réseaux sémantiques ne sont pas efficaces du point de vue de la communication avec l'utilisateur. Le schéma des concepts, encore augmenté de celui des occurrences, n'est plus utilisable comme outil de discussion.

Capacité de structuration des informations : Comme le schéma Entité/Association, le réseau sémantique met directement en relation les objets du monde réel avec leurs propriétés et leurs associations à d'autres objets. Il semble idéal pour donner l'accès à toutes les informations relatives à un concept donné,

et même à toutes les occurrences de ce concept. Pourtant, pour les yeux, les relations ne paraissent pas si évidentes, car le schéma n'est pas toujours très clair.

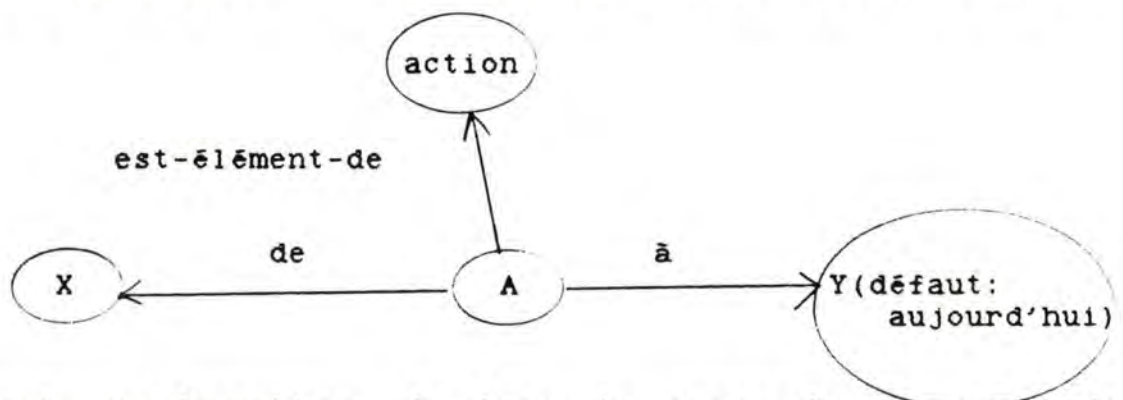
Expression de relations n-aires : L'expression de relations n-aires dans le formalisme des réseaux sémantiques nécessite, comme nous l'avons vu, certains artifices, qui peuvent paraître assez peu naturels, et qui obligent le lecteur à un effort supplémentaire d'interprétation.

Les traitements : Les réseaux sémantiques présentent une possibilité d'intervention des traitements dans leur propre formalisme, car si l'on exprime toutes les informations nécessaires, c'est-à-dire les objets manipulés, sous forme de noeuds et leur relations sous forme d'arcs, il est possible d'obtenir tous les résultats nécessaires à un traitement spécifique, en voyageant dans le réseau.

Les contraintes d'intégrité : Les contraintes d'intégrité seront également écrites comme les traitements.

Les valeurs par défaut : On trouve dans la littérature concernant les réseaux sémantiques des tentatives d'amélioration de leur contenu informatif, et notamment la prise en compte de valeurs par défaut au niveau des noeuds.

Par exemple, "l'action A s'est déroulée depuis la date X jusqu'à la date Y (par défaut, Y = aujourd'hui)" sera représentée par :



Cet ajout au formalisme de base le rapproche du mode de représentation suivant : les frames.

Les cas d'exception : Un réseau sémantique, puisqu'il parle en termes d'occurrences d'ensembles principalement, est idéal pour la représentation des cas d'exception. En effet, l'élément d'un ensemble hérite de toutes les propriétés de cet ensemble, par la

hiérarchie d'héritage de propriétés, mais il est possible aussi de lui attribuer des propriétés qui lui sont propres, qui constituent les cas d'exception dont il est question.

Le texte : Comme nous l'avons fait remarquer, la représentation du texte à l'état brut n'est certainement pas un atout des réseaux sémantiques, puisqu'ils ne disposent pas d'une structure suffisamment élaborée pour l'intégration du langage naturel.

Par contre, le codage d'un texte sous forme d'une grammaire semble pouvoir donner des résultats, ainsi qu'on a pu le voir à la section 11.4.1.3. Mais ce type d'analyse reste dans une optique de traduction automatique d'une langue dans une autre, ce qui n'est pas notre objectif.

Le temps : Le temps n'est pas non plus une notion privilégiée du réseau sémantique, bien que sa représentation soit spécifiée au même niveau que les autres concepts, c'est-à-dire sous forme de noeud, où aboutissent des arcs venant des différentes valeurs que prend le temps dans un contexte donné. Par rapport au modèle Entité/Association, dans lequel le temps peut être un simple attribut, il y a déjà une amélioration sensible. En effet, il suffirait de donner des propriétés sémantiques au concept de temps, pour que celles-ci soient héritées par tous les éléments temporels intervenant dans le schéma.

Les occurrences : Dans un domaine relatif à une taxonomie même complexe, le réseau sémantique est tout à fait indiqué pour la représentation des occurrences, puisque celles-ci sont les éléments d'ensembles dont les caractéristiques sont clairement définies, et auxquels on ajoute d'autres propriétés spécifiques. Mais lorsque l'on traite, comme ici, d'un cas où les relations sont beaucoup plus élaborées que les simples relations de sous-ensembles, d'instances ou de propriétés, la représentation des occurrences rend le schéma très compliqué et manquant surtout de structure.

Possibilité de raisonner directement sur le schéma : La règle d'inférence fondamentale pour le formalisme des réseaux est celle de l'héritage des propriétés. De plus, les systèmes de réseaux sémantiques utilisent la technique du matching, qui permet à l'utilisateur de raisonner sur des objets et des relations pour en déduire des propriétés inconnues auparavant. Bien manipulé, ce mécanisme donne la possibilité d'obtenir tout ce que l'on désire connaître à propos d'un objet donné. Bien sûr, il est nécessaire de connaître les relations présentes dans le schéma, ainsi que leur sémantique, ce qui constitue une limite dans l'interaction avec l'utilisateur.

Section 10.5. : Les frames et les scripts.

10.5.1. Définition des frames et des scripts.

10.5.1.1. Définitions.

Comme cela avait été suggéré dans la première section de ce chapitre, les "frames" sont des structures de représentation des connaissances incluant de l'information déclarative et procédurale.

Tout comme l'être humain utilise un corps de connaissances basé sur ses expériences précédentes, pour interpréter de nouvelles situations, les frames se servent des informations au sujet d'objets ou d'événements typiques, pour spécifier des états.

Un frame est une structure de données représentant un objet stéréotype ou une situation stéréotype.

Le niveau supérieur d'un frame contient son nom, qui l'identifie parmi les autres frames. Les niveaux inférieurs, appelés aussi éléments terminaux, ou "slots", sont des emplacements garnis avec des données relatives au concept représenté par le frame.

Les slots peuvent contenir des informations sur la façon d'utiliser le frame, sur les cas anormaux, sur les frames voisins, etc. C'est là aussi que se trouve l'information procédurale sous forme de fonctions ou de procédures à exécuter.

Voici un exemple de frame un peu élaboré :

MANAGER frame

spécialisation-de : EMPLOYE

nom :

 unité : (nom-famille, prénom, initiales)

âge :

 unité : années

adresse : ADRESSE

```

département :
    domaine : (production, administration)
salaire : SALAIRE
date-de-début :
    unité : (mois, année)
date-de-fin :
    unité : (mois, année)
défaut : maintenant

```

Nous avons ici un exemple typique de frame disposant de presque toutes les possibilités offertes par le formalisme.

MANAGER est le nom identifiant attribué à ce type d'entité précis. Le slot **spécialisation-de** correspond à la relation déjà étudiée dans les réseaux sémantiques : l'arc **est-un**, c'est-à-dire la relation de sous-ensemble, ou d'héritage de propriétés. Il va sans dire qu'elle est soumise aux mêmes problèmes sémantiques traités dans la section précédente.

Les mots **unité**, **domaine** et **défaut** sont ce que l'on appelle les facettes du slot. **Unité** exprime une restriction sur le type de la donnée contenue dans le slot, tandis que **domaine** donne la liste des valeurs parmi lesquelles le slot doit puiser son contenu. **Défaut** détermine une valeur par défaut, celle qui sera considérée en l'absence d'autre observation.

EMPLOYE, ADRESSE, SALAIRE sont des noms d'autres frames auxquels celui-ci s'associe pour mieux spécifier la description.

Il n'y a pas non plus de convention de syntaxe ni même de terminologie au sujet des facettes. Cela dépend du langage et du système de raisonnement utilisés.

L'exemple suivant permettra d'évaluer la capacité de traiter l'information procédurale dans un frame :

RESTAURANT frame

spécialisation de : ETABLISSEMENT-COMMERCIAL

type :

domaine : (fast-food, cafetaria, self-service,
service-aux-tables)

défaut : service-aux-tables

si-nécessaire :

SI comptoir-en-plastic-orange
ALORS fast-food

SI pile-de-plateaux
ALORS cafetaria

SI attente-de-la-serveuse OU réservation-faite
ALORS service-aux-tables
SINON self-service

lieu :

domaine : ADRESSE

si-nécessaire : (regarder dans le MENU)

Les facettes étiquetées par le terme **si-nécessaire** sont en fait des procédures à exécuter. Dans ce cas, on peut connaître la valeur du slot à partir de certaines données déjà connues qui ont été soit introduites, soit découvertes dans les autres slots. D'autres pourront entraîner une modification de la base de connaissances. Par exemple la facette :

si-ajout : (ajout lieu RESTAURANT)

implique un ajout du lieu du restaurant si celui-ci n'existe pas et doit être enregistré.

C'est une caractéristique très alléchante associée au formalisme des frames.

Cette structure élaborée va servir à créer et à maintenir des descriptions de concepts ou d'événements, grâce aux instantiations qui en seront déduites et qui signifieront les occurrences de l'objet étudié. Le slot **spécialisation-de** spécifie le frame-concept auquel appartient l'objet décrit, dont le nom est identifiant parmi tous les frames de la base de données. Les emplacements vides seront remplis selon les observations faites, ou les valeurs par défaut.

CHIEN frame

spécialisation-de : ANIMAL-DOMESTIQUE
élevage :
propriétaire :
domaine : PERSONNE

si-nécessaire : (Trouver une PERSONNE dont
l'animal est moi-même)

nom :
domaine : NOM-PROPRE
défaut : Eliot

a pour occurrence :

CHIEN-PRES-DE-LA-PORTE frame

spécialisation-de : CHIEN
élevage :
propriétaire : Jimmy
nom : Fido

10.5.1.2. Inférences.

Le mécanisme de déduction basé sur les frames se rapproche de la technique du "matching" dans les réseaux sémantiques :

Grâce à certaines caractéristiques connues (introduites au terminal, par exemple), un objet, ou une situation, est identifié par l'intermédiaire du frame qui paraît le mieux approprié. A partir des emplacements restés vides dans la question on peut déduire l'information manquante et la réponse au problème, en procédant parfois par héritage, par défaut, ou par procédures.

Si tous les slots ont pu être remplis, le frame est approprié pour comprendre la scène ou l'événement. Sinon, le contrôle est transféré à d'autres frames.

10.5.1.3. Critique générale.

Les constatations que nous pouvons retirer de cette analyse sont tout d'abord l'appréciation du caractère naturel de la représentation induite, facile à utiliser par les experts, ensuite et surtout, la capacité de manipulation d'une connaissance variée, statique ou dynamique, déclarative ou algorithmique.

-La structure interne des frames autorise les définitions par spécialisation et le maintien automatique des contraintes d'intégrité. Cette organisation de la connaissance peut être utile à la modularité et l'accessibilité de la connaissance. De

plus, les systèmes frames permettent de spécifier des valeurs par défaut pour des morceaux d'information sur un objet quand cette information n'est pas donnée explicitement.

Si l'on arrive à trouver des langages et des moteurs d'inférence capables de respecter la théorie des possibilités décrites ci-dessus, nous pouvons obtenir un système réellement puissant et efficace, car il raisonnera alors directement sur le modèle des connaissances tel qu'il est défini au niveau conceptuel.

Cette puissance est absente du modèle Entité/Association, puisqu'il nécessite une transformation en base de données relationnelles ou réseaux, dans l'état actuel des technologies.

Il est à noter que des chercheurs tentent de faire du langage fonctionnel LISP le langage d'implémentation des frames, et qu'ils aboutissent à certains résultats [].

Les frames peuvent être intégrés aux règles de production pour former une technique de représentation unique [NAU]. Un des avantages majeurs de ce type de représentation hybride est de rendre le pouvoir d'organisation et d'expression de la programmation orientée-objet disponible aux experts non programmeurs.

10.5.1.4. Les scripts.

Le formalisme des "scripts" que nous annonçons dans la section introductive est un dérivé des frames.

Les scripts décrivent plus précisément des scénarios stéréotypés, des suites d'actions, des séquences d'événements mettant en cause des personnages. Ils s'intéressent, plus encore que les frames, à l'aspect dynamique de la connaissance.

Leur but est de compléter une chaîne causale reliant les actions décrites et les données introduites. Comme les frames, ils peuvent aussi faire appel à d'autres scripts pour les besoins des inférences.

Ces deux techniques sont, en résumé, des méthodes d'organisation de la connaissance destinées à diriger l'attention et à faciliter l'accès et l'inférence.

10.5.2. Application des frames au modèle juridique.

10.5.2.1. Schéma conceptuel.

Dans le modèle qui suit, nous avons conservé en tant qu'entités distinctes, c'est-à-dire en tant que frames, les concepts de STANDARD, FAIT, CLASSE, CONTEXTE-NECESSAIRE, CONTEXTE-SPECIFIQUE, JUGEMENT, COMPOSITION, FAIT-PONDERE, STANDARD-PONDERE ET PERSONNE. Chacun d'eux possède comme slots les attributs qui le qualifiaient dans le schéma Entité/Association, ainsi que des emplacements réservés aux associations avec d'autres frames.

Par exemple, dans le frame STANDARD, il existe un emplacement réservé à son contexte d'application, sous forme d'un ou plusieurs frames CONTEXTE-NECESSAIRE, associés éventuellement à un ou plusieurs CONTEXTE-SPECIFIQUE, ou sous forme d'un ou plusieurs CONTEXTE-SPECIFIQUE seulement. Le concept de date a été transformé en une entité frame séparée, car il subit des traitements spécifiques.

SCHEMA OBTENU :

STANDARD frame

```
n-identifiant :
    unité : entier
    cardinalité-min : 1
texte :
    unité : chaîne de caractères
polarisation :
    domaine : (+, -)
état :
    domaine : (affirmé, nié)
    défaut : affirmé
n-standard-opposé :
    unité : entier
    défaut : 0
contexte-d'application :
    domaine : (CONTEXTE-NECESSAIRE,
              (CONTEXTE-NECESSAIRE & CONTEXTE-SPECIFIQUE),
              CONTEXTE-SPECIFIQUE)
    cardinalité-max : ∞
```


FAIT frame

n.identifiant :
 unité : entier
 cardinalité-min : 1
texte :
 unité : chaîne de caractères
polarisation :
 domaine : (+, -)
texte-nié : chaîne de caractères
classe : CLASSE

CLASSE frame

n.identifiant :
 unité : entier
 cardinalité-min : 1
libellé :
 unité : chaîne de caractères

CONTEXTE-NECESSAIRE frame

n.identifiant :
 unité : entier
 cardinalité-min : 1
date : DATE
conditions-nécessaires-faits :
 unité : (FAIT, domaine : (affirmé, nié), domaine : (loi, doctrine, cour de cassation))
 cardinalité-min : SI cardinalité de conditions-nécessaires-standards = 0
 ALORS 1
 SINON 0
 cardinalité-max : ∞
conditions-nécessaires-standards :
 unité : (STANDARD, domaine : (loi, doctrine, cour de cassation))
 cardinalité-min : SI cardinalité de conditions-nécessaires-faits = 0
 ALORS 1
 SINON 0
 cardinalité-max : ∞
valide :
 domaine : JUGEMENT
 cardinalité-max : ∞

CONTEXTE-SPECIFIQUE frame

n.identifiant :
 unité : entier
 cardinalité-min : 1
conditions-spécifiques-faits :
 unité : (FAIT, domaine : (affirmé,nié))
 cardinalité-min : SI cardinalité de
 conditions-spécifiques-standards = 0
 ALORS 1
 SINON 0
 cardinalité-max : ∞
conditions-spécifiques-standards :
 domaine : STANDARD
 cardinalité-min : SI cardinalité de
 conditions-spécifiques-faits = 0
 ALORS 1
 SINON 0
 cardinalité-max : ∞
dans :
 domaine : JUGEMENT
 cardinalité-min : 1
 cardinalité-max : ∞

JUGEMENT frame

n.identifiant :
 unité : entier
 cardinalité-max : ∞
date : DATE
lieu :
 unité : chaîne de caractères
type :
 domaine : (cassation, second degré, premier degré)
résultat :
 domaine : (libération, maintien)
casse :
 domaine : (oui, non)
texte :
 unité : chaîne de caractères

DATE frame

année :
 unité : entier de 4 chiffres
mois :
 domaine : [1..12] ou (janvier,...,décembre)
jour :

domaine : SI mois = janvier, mars, mai, juillet, août,
octobre ou décembre,

ALORS [1..31]
SI mois = avril, juin, septembre, novembre,
ALORS [1..30]
SI mois = février et année non bissextile
ALORS [1..28]
SINON [1..29]

COMPOSITION frame

antécédent :
domaine : (FAIT-PONDERE, STANDARD-PONDERE, COMPOSITION)
cardinalité-min : 1
cardinalité-max : ∞
conséquent :
domaine : (FAIT-PONDERE, STANDARD-PONDERE, COMPOSITION)
cardinalité-min : 1
négation :
domaine : (oui, non)
effet :
domaine : (libération, maintien)
jugement :
domaine : JUGEMENT
cardinalité-min : 1
cardinalité-max : ∞

FAIT-PONDERE frame

fait : FAIT
état :
domaine : (affirmé ou nié)
pondération :
domaine : (importante, subsidiaire, nulle)

STANDARD-PONDERE frame

standard : STANDARD
pondération :
domaine : (importante, subsidiaire, nulle)

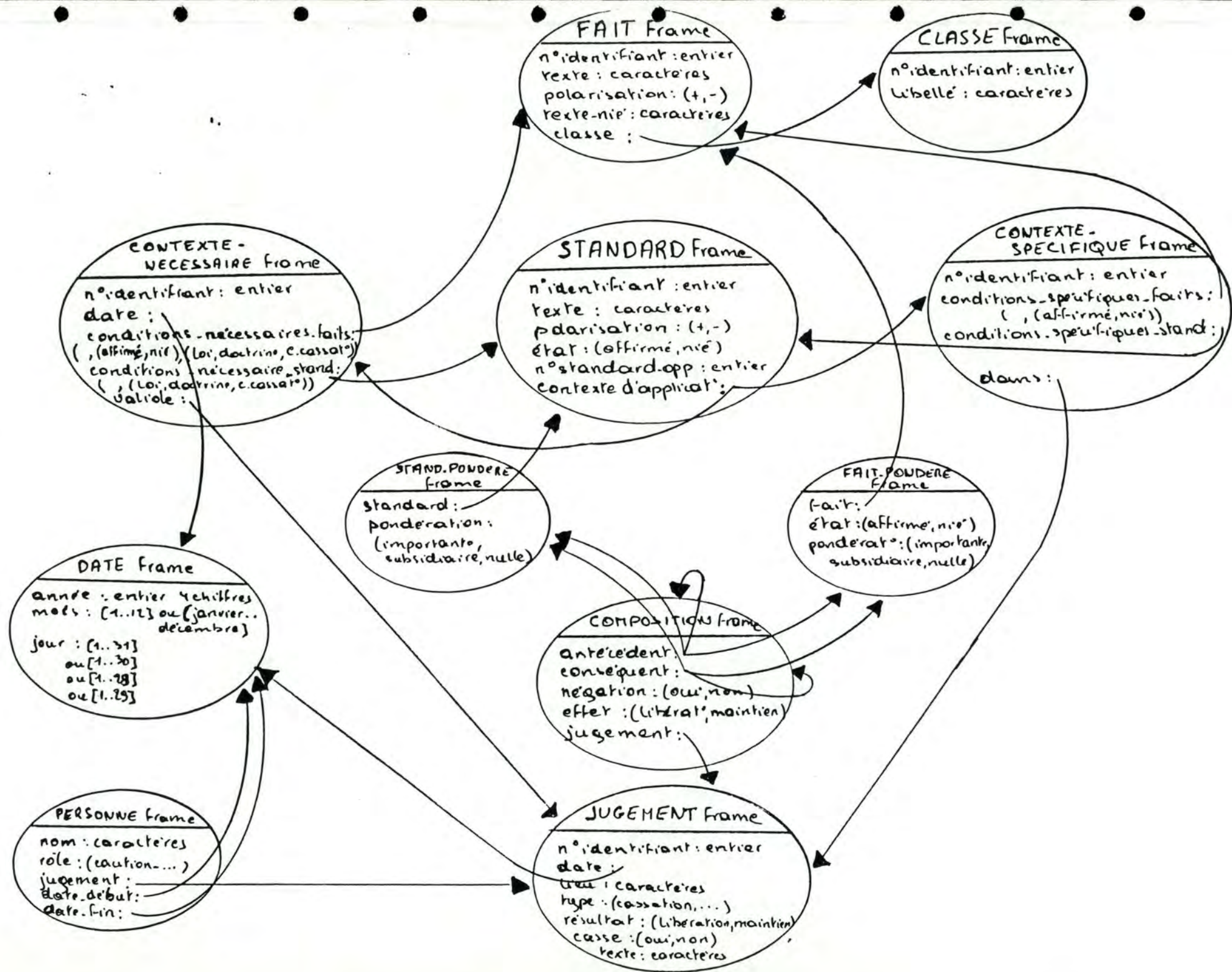
PERSONNE frame

nom :
unité : chaîne de caractères
rôle :
domaine : (caution-nouvelle, caution-originale,
débiteur-principal, créancier, nouvelle-forme)

jugement : JUGEMENT
date-début-rôle : DATE
date-fin-rôle :
 domaine : DATE
 défaut : aujourd'hui

Lorsqu'elle n'est pas spécifiée, nous conviendrons que la cardinalité du contenu des slots est au minimum de 0 et au maximum de 1.

Une autre représentation des frames, plus proche de celle d'un réseau sémantique, permet d'éclaircir leurs interrelations :



10.5.2.2. Raisonnement.

Comment raisonner sur le schéma obtenu, pour répondre aux fonctions du prototype ?

Soit la fonction qui donne le(s) standard(s) à appliquer d'après certains identifiants de faits introduits et une polarisation.

Chaque numéro de fait sera introduit dans le slot n-identifiant d'un frame FAIT non encore instancié. Celui-ci sera "matché" avec l'occurrence correspondante d'un frame FAIT.

Les frames retenus tenteront de matcher avec l'ensemble des FAITs contenus dans le slot **conditions-nécessaires-faits** d'un frame CONTEXTE-NECESSAIRE et/ou dans le slot **conditions-spécifiques-faits** d'un frame CONTEXTE-SPECIFIQUE.

Si l'opération réussit, nous obtenons les autres renseignements concernant le contexte, et par le même mécanisme, le frame STANDARD dont il régit l'application, si celui-ci possède la polarisation donnée en entrée.

Si nous voulons connaître le(s) contexte(s) d'application d'un standard donné, il suffit de matcher son numéro avec le n-identifiant d'un frame STANDARD, dont on retire alors les informations nécessaires, à savoir les frames CONTEXTE-NECESSAIRE et CONTEXTE-SPECIFIQUE associés.

Il est donc possible de collecter les informations d'une base de données frames de plusieurs façons, sans en modifier la représentation.

Le grand avantage de ce formalisme, en plus de sa clarté évidente, est de pouvoir parler directement en termes d'entités, de concepts, ici représentés par des frames, aussi bien qu'en termes d'occurrences, qui sont des frames dont la structure correspond à celle du frame-concept dont elles sont la spécialisation. Pour citer un frame, le nom suffit, et il n'est pas nécessaire de lui donner un autre attribut identifiant.

10.5.2.3. Mises à jour.

Cependant, dans ce schéma, certaines informations manquent pour pouvoir effectuer une mise à jour de la base. Pour ce faire, à l'intérieur des frames susceptibles d'être concernées par une modification, il suffit d'ajouter les facettes **si-ajout**, **si-retrait**, qui seront les conditions d'exécution de la procédure qui les suit.

Exemple : Mise à jour des contextes d'application d'un standard :

STANDARD frame

```
n.identifiant :
    unité : entier
    cardinalité-min : 1
texte :
    unité : chaîne de caractères
polarisation :
    domaine : (+, -)
état :
    domaine : (affirmé, nié)
    défaut : affirmé
n.standard-opposé :
    unité : entier
    défaut : 0
contexte-d'application :
    domaine : (CONTEXTE-NECESSAIRE,
              (CONTEXTE-NECESSAIRE & CONTEXTE-SPECIFIQUE),
              CONTEXTE-SPECIFIQUE)
    cardinalité-max : ∞
    si-ajout : (ajout contexte (n.identifiant))
    si-retrait : (retrait contexte (n.identifiant))
```

Mise à jour des conditions d'application d'un contexte:

CONTEXTE-NECESSAIRE frame

```
n.identifiant :
    unité : entier
    cardinalité-min : 1
date : DATE
conditions-nécessaires-faits :
    unité : (FAIT, domaine : (affirmé, nié), domaine : (loi,
    doctrine, cour de cassation))
```

```

cardinalité-min : SI cardinalité de
                    conditions-nécessaires-standards = 0
                    ALORS 1
                    SINON 0
cardinalité-max : ∞
si-ajout : (ajout cond-nec-faits (n·identifiant))
si-retrait : (retrait cond-nec-faits (n·identifiant))
conditions-nécessaires-standards :
    unité : (STANDARD, domaine : (loi, doctrine, cour de
        cassation))
cardinalité-min : SI cardinalité de
                    conditions-nécessaires-faits = 0
                    ALORS 1
                    SINON 0
cardinalité-max : ∞
si-ajout : (ajout cond-nec-stands (n·identifiant))
si-retrait : (retrait cond-nec-stands (n·identifiant))
valide :
    domaine : JUGEMENT
    cardinalité-max : ∞

```

Voici donc le fameux "attachement procédural" qui fait des frames un formalisme de représentation mixte, mêlant les connaissances déclaratives et les programmes, ce qu'aucune autre technique vue précédemment ne proposait de manière aussi organisée et structurée.

10.5.2.4. Conclusion.

Bien entendu, ainsi que nous l'avons souligné plus haut, l'utilité de cette forme de représentation va dépendre très fort de l'implémentation sous-jacente, qui devrait permettre de profiter au maximum de la structure de propriétés, du contrôle des cardinalités, du matching, des inférences et des procédures.

10. 5. 3. Comparaison des frames avec le modèle Entité/Association.

Les frames sont en réalité des réseaux sémantiques améliorés. Par conséquent, ils conservent leurs avantages en limitant leurs inconvénients :

Simplicité du formalisme : Les frames ajoutent aux réseaux sémantiques les notions de slots et de facettes. Mais leur formalisme reste simple, pourvu que le langage dans lequel ils sont implémentés le soit aussi. En effet, le contenu des facettes varie selon la syntaxe autorisée : représentation des types, des domaines de valeurs, des procédures, de leurs arguments, etc. Excepté ces problèmes de représentation du contenu des slots, qui peut, conceptuellement, être rédigé en langage naturel, les trois seuls éléments à retenir sont les frames, les slots et les facettes.

Visualisation et communication avec l'utilisateur : Sous leur forme première, les frames ne donnent pas une vue d'ensemble du sujet, bien que leur signification soit claire et précise. Par contre, leur représentation sous forme de réseau mettant en évidence leurs interrelations, est sans aucun doute plus lisible, et se rapproche très fort de celle du modèle E/A, sans les associations n-aires ($n > 2$). Le fait que les propriétés directement liées aux concepts soient reprises à l'intérieur du même cercle donne une formalisation plus claire que celle des réseaux sémantiques.

Capacité de structuration des informations : Ainsi que le schéma E/A et les réseaux sémantiques, les frames permettent l'accès instantané aux informations relatives à un objet donné.

Expression de relations n-aires : Le même artifice que pour les réseaux sémantiques peut être utilisé par les frames, mais il implique les mêmes problèmes de visualisation. Ce formalisme n'est donc pas réellement adapté à la représentation des relations n-aires, mais bien à celle des relations binaires en général.

Extensibilité : La remarque valable pour les réseaux sémantiques l'est aussi dans le cas des frames : aucun problème d'extensibilité.

Les traitements : Ce critère constitue le plus gros avantage des frames par rapport aux réseaux sémantiques. En effet, des procédures de traitement peuvent sans problèmes être introduites

dans le contenu des facettes, entraînant ainsi l'exécution de fonctions décrites par ailleurs. De cette façon, les frames peuvent s'auto-gérer, c'est-à-dire déduire leurs propres valeurs d'éléments de données extérieurs ou se modifier eux-mêmes. Les procédures sont partie intégrante du formalisme, et ont de plus la capacité d'être exécutées !

Les contraintes d'intégrité : Les contraintes d'intégrité peuvent être écrites dans le contenu des facettes, selon la syntaxe appropriée au langage. A tout moment, le système vérifie si elles sont bien respectées.

Les valeurs par défaut : Cette possibilité fait partie du formalisme, comme nous l'avons constaté à la section 11.5.1.

Les cas d'exception : Les cas d'exception sont traités par les frames comme par les réseaux sémantiques, puisqu'ils disposent des mêmes relations d'héritage de propriétés. Le contenu d'un slot peut servir à donner une valeur particulière lorsque certaines conditions d'exception sont présentes.

Le texte : De nouveau, le langage sous-jacent aux frames va déterminer la possibilité de représenter le langage naturel, mais il n'y aurait vraisemblablement aucune contre-indication au fait que le texte soit seulement une chaîne de caractères dans un slot.

Le temps : Le concept de temps peut être traité à l'intérieur d'un frame spécial, et faire l'objet de valeurs ou de vérifications particulières, selon les désirs du concepteur.

Les occurrences : Les frames représentant les occurrences de frames génériques y sont reliés par la relation **spécialisation-de**, qui leur transmet les mêmes propriétés. Les éléments d'ensembles s'inscrivent donc dans un cadre précis, contrairement aux réseaux sémantiques, qui ne possèdent pas réellement une structure enfermant les caractéristiques d'un objet. La représentation et l'utilisation des occurrences permet aux frames de déduire des valeurs pour des objets particuliers, et d'effectuer des traitements les concernant.

Possibilité de raisonner directement sur le schéma : L'héritage de propriétés, le matching et les procédures de traitement sont les seuls outils nécessaires aux inférences sur base des connaissances formalisées sous forme de frames. Comme les réseaux sémantiques, les frames sont un langage orienté-objet, sur lequel on peut raisonner pour déduire des réponses aux questions que l'on se pose. Certaines fonctions prédéfinies peuvent également être exécutées, puisqu'il suffit alors de faire appel au frame qui les contient.

C'est ce dernier point qui fait toute la puissance du formalisme des frames par rapport aux autres modes de représentation étudiés.

Section 10.6. : Conclusion.

Tous les formalismes étudiés dans le Chapitre 11 se distinguent du modèle Entité/Association en ce qu'ils sont implémentables et manipulables sous leur forme première par l'utilisateur final.

En effet, ils disposent chacun d'un mécanisme de raisonnement qui leur est associé, et qui permet de déduire directement les résultats demandés.

La seconde différence importante, liée à la première, est que le schéma Entité/Association est fait pour représenter des concepts et non des occurrences. Celles-ci ne seront introduites qu'après plusieurs étapes de transformation du schéma E/A en une base de données et après programmation.

Pourtant, ces autres formalismes, particulièrement la logique, les réseaux sémantiques et les frames, peuvent servir à remplacer le schéma E/A dans l'étape de représentation conceptuelle des données, tout en présentant l'avantage supplémentaire de ne plus demander par la suite un effort de programmation. L'inconvénient qui en découle, surtout dans le cas des réseaux sémantiques, est que les occurrences ne doivent pas nécessairement respecter le schéma des concepts.

Du point de vue de la clarté et de la communicabilité avec l'expert, ainsi que pour l'expression de relations n-aires, le schéma E/A conserve certains avantages par rapport aux autres formalismes.

Par contre, les frames sont les seuls à remplir parfaitement les critères de représentation des traitements, des contraintes d'intégrité, des valeurs par défaut, des cas d'exception, du texte et même du temps. Leur formalisme est relativement simple, et ils n'ont rien à envier aux réseaux sémantiques, au calcul des prédicats ou aux règles de production en ce qui concerne l'extensibilité et la disponibilité des informations.

Dans notre cas, il serait donc très intéressant d'approfondir cette technique de représentation des connaissances, car elle apporte beaucoup plus que le simple schéma Entité/Association, surtout pour les traitements, tout en ne perdant pas tellement au niveau de la clarté.

CONCLUSION.

L'objectif de ce mémoire était de revoir le modèle de la décision judiciaire, de façon à étendre le prototype réalisé dans le mémoire précédent.

Après avoir étudié la décision judiciaire de façon théorique, nous avons finalement élaboré un nouveau modèle, celui de l'argumentation juridique, qui nous a permis de développer de nouvelles fonctions, très proches du raisonnement habituel de l'avocat ou du juge. En effet, la mémorisation des contextes d'application des normes juridiques répond à la démarche habituelle du juriste disposant d'un cas d'espèce, et désirant argumenter son point de vue.

L'extension du modèle juridique nous a donné l'occasion de relever et de critiquer quelques formalismes de représentation des connaissances, et de proposer l'un d'eux, les "frames", comme technique de représentation du jugement et des conditions d'application des normes.

Malheureusement, le temps nous a manqué pour approfondir l'étude d'un langage d'implémentation des frames. Il serait fort intéressant de continuer dans cette voie, car ce formalisme paraît très puissant, et bien adapté aux connaissances juridiques. Il permettrait probablement d'intégrer en un seul système les deux prototypes actuellement distincts : le prototype documentaire et d'évaluation, et celui de l'argumentation.

Le manque d'expérimentation par les juristes concernés reste encore une limite importante du travail, et les quelques propositions d'extensions citées au Chapitre 2 (extension des requêtes, extension des tests, réalisation d'une interface système documentaire - système d'évaluation) restent évidemment sans suite à l'heure actuelle.

Plus que jamais, des contacts avec d'autres centres de recherche en informatique et droit peuvent apporter des idées neuves concernant la structuration du jugement, et les besoins des utilisateurs.

Les pistes de recherche restent donc nombreuses, tant au niveau technique que conceptuel.

BIBLIOGRAPHIE.

[BARR 81] A. BARR, E. FEIGENBAUM, Handbook of Artificial Intelligence, Vol. 2, Ch. 3 : Representation of Knowledge, Los Angeles, 1981-1982.

[BAUE 85] H. BAUER-BERNET, Les aspects temporels dans la formalisation du droit, Proceedings "Logica, Informatica, Dirrito", Firenze 1985.

[BIM 85] Six-monthly progress report for the period february to july 1985, ed. V-Ostler, Project technical coord., Scicon limited.

[BODA 85] F. BODART, M. HELLA, Y. POULLET, P. STENNE, Un prototype de système d'information d'aide à l'élaboration de jugements, Centre de Recherche Informatique et Droit, F. U. N. D. P., Namur, 1985.

[BODA 83] F. BODART, Y. PIGNEUR, Conception assistée des applications informatiques, 1. Etude d'opportunité et analyse conceptuelle, Masson, Presses Universitaires de Namur, 1983.

[BULY 85] E. BULYGIN, Permissive Norms and Normative Systems, Proceedings "Logica, Informatica, Dirrito", Firenze 1985.

[CLOC 81] W. F. CLOCKSIM, C. S. MELLISH, Programming in Prolog, SPRINGER-VERLAG, 1981.

[COMP] Computational approaches to representation and control, Expert systems - Concepts and examples, Ch. 3,

[DBAS] DBASE2 - USER MANAUL, ASHTON-TATE.

- [DOLK 84] D. R. DOLK, B. R. KONSINSKI, Knowledge representation for model management systems, 1984.

[DOMA] P. DOMANGE, E. MEIRLAEN, CML, a conceptual modelling language for software engineering.

[DOPP 65] J. DOPP, Notions de logique formelle, B. Nauwelaerts, Paris, 1965.

[FIKE 85] R. FIKES, T. KELLER, The role of frame-based representation in reasoning, Communications of the ACM, September 85, Vol. 28, Numb. 9.

[GALL 84] H. GALLAIRE, Systèmes experts, Techniques de l'ingénieur, 9, 1984.

[GENE 85] M. R. GENESERETH, M. L. GINSBERG, Logic programming, Communications of the ACM, September 85, Vol. 28, Numb. 9.

[HAYE 85] F. HAYES-ROTH, Rule-based systems, Communications of the ACM, September 85, Vol. 28, Numb. 9.

[HICK 82] B. HICK, P. TIRTIAUX, Système informatique d'aide au jugement de qualification, Mémoire de l'Institut d'Informatique, F. U. N. D. P., Namur, 1982.

[JARK 84] M. JARKE, J. VASSILIOU, Coupling Expert Systems with Database Management Systems, Artificial Intelligence Applications for Business, W. Reitman (ed.), Ablex, 1984.

[KLOP 85] M. R. KLOPPROGE, TERM : An approach to include the Time dimension in the Entity/Relationship Model, Proceedings 2nd Inter. Conf. on E. R. Approach, Washington, 1981.

[KOWA 83] R. KOWALSKI, Logic programming, Proceedings of the 9th IFIP World Computer Congress, 19-23, september 83.

[KUIP 75] B. J. KUIPERS, A frame for frames : Representing knowledge for recognition, Representation and understanding, D. G. BOBROW, p 151-184, Academic press, 1975.

[LAUR 81] J-L. LAURIERE, Les systèmes experts, TSI-AFCET (F) 1 n°1, 1981.

[LAUR 81] J-L. LAURIERE, Représentation et utilisation des connaissances, Techniques et Science Informatique, Vol.1, N°2, 1982.

[LEAR] LEARNCALC : een op logica gebaseerd veralgemeend elektronisch spreadsheet dat bijleert, BIM, Departement Onderzoek, KUL, Departement Computer wetenschappen.

[MATU 85] R. MATUSZEWSKI, MIZAR-MSE, Enseignement des fondements de la mathématique appuyé par ordinateurs, Août 1985.

[McCAB] F.G. McCABE, K.L. CLARCK, B.D. STEEL, MICRO-PROLOG (A IBM-PC Version), Programmer's Reference Manual.

[McCA 85] L.T. McCARTY, Permissions and obligations, Proceedings "Logica, Informatica, Dirrito", Firenze 1985.

[McDE 78] J. Mc DERMOTT AND AL., Production system conflict resolution strategies, pp 177-199, D.A. WATERMAN, HAYES-ROTH, Pattern-directed inference systems, Academic Press, 1978.

[MIZA] MIZAR-MSF implemented on IBM PC/XT (PC-DOS), Technical report.

[MOST 85] M. MOSTOWSKI, N. ZALEWSKA, Logical exercises in the MIZAR-MSE language, Fourdrain, 1985.

[NAU 83] D.S. NAU, University of Maryland, Special Feature : Expert Computer Systems, Feb. 1983.

[PINS 81] S. PINSON, Représentation des connaissances dans les systèmes experts, RAIRO Informatique / Computer Science, Vol.15, n°4, 1981, p. 343 à 367).

[PERE] F. PEREIRA, CProlog User's Manual, SRI International, Menlo Park, California.

[PONC 84] I. PONCIN, P. STENNE, Réalisation d'un prototype d'un système informatique d'aide à l'élaboration d'un jugement, Mémoire de l'Institut d'Informatique, F.U.N.D.P., Namur, 1984.

[PRAZ 83] K. PRAZMOWSKI, P. RUDNICKI, A draft of MIZAR-MSE primer, Warsaw, June 1983.

[PROC 85] Proceedings "Logica Informatica Dirritto", Firenze, 1985.

[RAPP] Rapport CadisII.

[SANC 85] D. SANCHEZ GARCIA, Logique déontique et droit naturel, Proceedings "Logica, Informatica, Dirritto", Firenze 1985.

[SCHA 86] M. SCHAUSS, Propositions relatives à un prototype d'élaboration d'argumentations, C.R.I.D, F.U.N.D.P., Namur, 1986.

[TARS 71] A. TARSKI, Introduction à la logique, Gauthier. Villars, Paris, 1971.

[TRYB] A. TRYBULEC, S. ZUKOWSKI, Advanced course of MIZAR-MSE.

[VANA 85] F. VAN ASSCHE, Expert Systems, Knowledge bases and databases, Control Data Belgium, Brussel.

[VANH 85] P. VANHENTENRYCK, Logique et bases de données, Mémoire de l'Institut d'Informatique, F.U.N.D.P., Namur, 1985.

[VASS 85] T. VASSILIOU, J. CLIFFORD, M. JARKE, Acces to specific declarative knowledge by E.S. : The impact of logic programming, Elsevier Science Publisher B.V., 1985.

[VENK] R. VENKEN, BIMProlog, B.I.M., Everberg.

[WINS 84] WINSTON, Logic and theorem proving, Artificial Intelligence, 2nd ED., Ch. 7, Addison Wesley, 1984.

[WRO 72] J.WROBLEWSKI, Computers as an aid to the judicial process, Systema I, 1972.

[WROB 73] J.WROBLEWSKI, Archiv. F.Rechts - Und Sozial - philosophie 2, 1973.

[WROB 82] J.WROBLEWSKI, L'interprétation en droit : Théorie et idéologie, Archives de philosophie du droit, XVII, 1972.

[WROB 79] J.WROBLEWSKI, Justification of legal decisions, Revue internationale de philosophie, 127-128, 1979.

[WROB 69] J.WROBLEWSKI, Legal reasoning in legal interpretation, Logique et Analyse, 1969.

[WROB 78] J.WROBLEWSKI, Motivation de la décision judiciaire, La motivation des décisions de justice, Bruxelles, 1978.

[WRO 81] J.WROBLEWSKI, Operative models and legal systems, Proceedings "Logica, Informatica, Dirrito", Firenze 1981.

[WROB 70] J.WROBLEWSKI, An outline of the principal problems of the relations between law and cybernetics, Centro di Giuscibernetica dell'Universitica de Torino, Bulletin 2, Janvier-Mars, 1970.

[WROB 81] J.WROBLEWSKI, La preuve juridique : Axiologie, Logique et argumentation, La preuve en droit, Bruxelles 1981.

[WROB 85] J.WROBLEWSKI, Representational Models of legal systems, Proceedings "Logica, Informatica, Dirrito", Firenze 1985.

[WROB 84] J.WROBLEWSKI, Le rôle des principes du droit dans la théorie et l'idéologie de l'interprétation juridique, Archivum -Iuridicum Cracoviense XVII, 1984.

[WROB 82] J.WROBLEWSKI, Three concepts of validity of law,
Tidskrift, 5-6, 1982.

[WROB 83] J.WROBLEWSKI, Towards foundations of judicial
reasoning, Metatheorie juristischer Argumentation, Berlin, 1983.

A>prolog

micro-PROLOG 3.1 - created 29 Mar 84
(c) 1984 Logic Programming Associates Ltd.
62770 + 16384 Bytes Free

&.
LOAD TRTCONT
&7((trtcont))

Désirez-vous

1. Consulter les contextes d'un standard.
2. Connaître la meilleure argumentation à défendre pour une polarisation donnée.
3. Connaître la meilleure argumentation à défendre pour des faits donnés (et une polarisation donnée).
4. Mettre à jour les contextes d'un standard.
9. Retour au menu précédent.

Votre choix : 1

Introduisez le numéro du standard dont vous désirez connaître
le contexte : 4

Ce standard n'a aucune condition nécessaire à son application

CONTEXTE SPECIFIQUE 5 :

Ce contexte a été utilisé dans le(s) jugement(s) (2) :

Le fait 1 à l'état AFFIRME
->

CONTEXTE SPECIFIQUE 6 :

Ce contexte a été utilisé dans le(s) jugement(s) (3) :

Le fait 3 à l'état AFFIRME
->

Le standard 2
->

Désirez-vous connaître les contextes d'application
de ce standard 2 ? (y/n) y

CONTEXTES DU STANDARD 2 :

CONTEXTE NECESSAIRE 3 défini le 1 1 1980 :

Le fait 1 à l'état AFFIRME défini par la CASSATION
->

CONTEXTE SPECIFIQUE 3 associé au contexte 3 :

Ce contexte a été utilisé dans le(s) Jugement(s) (1) :

Le fait 2 à l'état NIE
->

Le fait 3 à l'état AFFIRME
->

Il n'existe pas ou plus de contexte d'application
du standard 2 .

Suite de la liste des contextes du standard 4 :

Il n'existe pas ou plus de contexte spécifique
d'application du standard 4 .

Désirez-vous introduire un autre standard ?(y/n)y

Introduisez le numero du standard dont vous désirez connaître
le contexte : 1

CONTEXTES DU STANDARD 1 :

CONTEXTE NECESSAIRE 1 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la LOI
->

Le fait 3 à l'état NIE défini par la LOI
->

Le standard 3 défini par la LOI
->

Désirez-vous connaître les contextes d'application
de ce standard 3 ? (y/n) y

CONTEXTES DU STANDARD 3 :

CONTEXTE NECESSAIRE 4 défini le 1 1 1980 :

Le fait 3 à l'état NIE défini par la LOI
->

CONTEXTE NECESSAIRE 5 défini le 1 1 1980 :

Le fait 1 à l'état NIE défini par la LOI
->

CONTEXTE SPECIFIQUE 4 associé au contexte 5 :

Ce contexte a été utilisé dans le(s) jugement(s) (1) :

Le fait 3 à l'état NIE

CONTEXTE NECESSAIRE 6 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la CASSATION
->

Le fait 1 à l'état NIE défini par la CASSATION
->

Il n'existe pas ou plus de contexte d'application
du standard 3 .

Suite de la liste des contextes du standard 1 :

CONTEXTE SPECIFIQUE 1 associé au contexte 1 :

Ce contexte a été utilisé dans le(s) jugement(s) (1) :

Le fait 1 à l'état NIE
->

CONTEXTE SPECIFIQUE 2 associé au contexte 1 :

Ce contexte a été utilisé dans le(s) jugement(s) (2) :

Le fait 1 à l'état NIE
->

Le standard 5
->

Désirez-vous connaître les contextes d'application
de ce standard 5 ?(y/n)n

CONTEXTE NECESSAIRE 2 défini le 1 1 1980 :

Le fait 1 à l'état NIE défini par la LOI
->

Le fait 2 à l'état AFFIRME défini par la CASSATION
->

Il n'existe pas ou plus de contexte d'application
du standard 1 .

Désirez-vous introduire un autre standard ?(y/n)n

Désirez-vous

1. Consulter les contextes d'un standard.
2. Connaître la meilleure argumentation à défendre pour une polarisation donnée.
3. Connaître la meilleure argumentation à défendre pour des faits donnés (et une polarisation donnée).
4. Mettre à jour les contextes d'un standard.
9. Retour au menu précédent.

Votre choix : 2

Introduisez une polarisation + ou - : +

Désirez-vous obtenir les standards et leurs contextes
dans un ordre de préférence?(y/n)y

Les standards à utiliser sont, par ordre de préférence :

LE STANDARD 2 :

en utilisant le CONTEXTE NECESSAIRE 3 défini le 1 1 1980 :

Le fait 1 à l'état AFFIRME défini par la CASSATION
->

CONTEXTE SPECIFIQUE 3 associe au contexte 3 :

Ce contexte a été utilisé dans le(s) jugement(s) (1) :

Le fait 2 à l'état NIE
->

Le fait 3 à l'état AFFIRME
->

LE STANDARD 4 :

Ce standard n'a aucune condition nécessaire à son application

CONTEXTE SPECIFIQUE 6 :

Ce contexte a été utilisé dans le(s) jugement(s) (3) :

Le fait 3 à l'état AFFIRME

->

Le standard 2

->

Désirez-vous connaître les contextes d'application
de ce standard 2 ? (y/n) n

LE STANDARD 4 :

Ce standard n'a aucune condition nécessaire à son application

CONTEXTE SPECIFIQUE 5 :

Ce contexte a été utilisé dans le(s) jugement(s) (2) :

Le fait 1 à l'état AFFIRME

->

Désirez-vous

1. Consulter les contextes d'un standard.
2. Connaître la meilleure argumentation à défendre pour une polarisation donnée.
3. Connaître la meilleure argumentation à défendre pour des faits donnés (et une polarisation donnée).
4. Mettre à jour les contextes d'un standard.
9. Retour au menu précédent.

Votre choix : 3

Introduisez les faits sous la forme
(((n°fait) <AFFIRME ou NIE>) ((n°fait) <AFFIRME ou NIE>))etc.
((1 NIE) (2 AFFIRME))

Introduisez une polarisation sous la forme + ou -
(0 si pas d'effet particulier souhaité) : +

Désirez-vous obtenir les standards et leurs contextes applicables dans
de préférence?(y/n)y

Les standards applicables sont, par ordre de préférence :

LE STANDARD 3 :

en utilisant le CONTEXTE NECESSAIRE 5 défini le 1 1 1980 :

Le fait 1 à l'état NIE défini par la LOI

->

LE STANDARD 1 :

en utilisant le CONTEXTE NECESSAIRE 2 défini le 1 1 1980 :

Le fait 1 à l'état NIE défini par la LOI

->

Le fait 2 à l'état AFFIRME défini par la CASSATION

->

LE STANDARD 3 :

en utilisant le CONTEXTE NECESSAIRE 6 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la CASSATION

->

Le fait 1 à l'état NIE défini par la CASSATION

->

LE STANDARD 5 :

en utilisant le CONTEXTE NECESSAIRE 7 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la LOI

->

Désirez-vous connaître un fait supplémentaire qui, s'il était prouvé, permettrait d'utiliser d'autres contextes?(y/n)y

Si vous pouvez prouver le FAIT SUPPLEMENTAIRE 3 NIE ,

les standards suivants seront également applicables :

LE STANDARD 3 :

en utilisant le CONTEXTE NECESSAIRE 4 défini le 1 1 1980 :

->

LE STANDARD 1 :

en utilisant le CONTEXTE NECESSAIRE 1 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la LOI
->

Le fait 3 à l'état NIE défini par la LOI
->

Le standard 3 défini par la LOI
->

Désirez-vous connaître les contextes d'application
de ce standard 3 ?(y/n)n

CONTEXTE SPECIFIQUE 2 associe au contexte 1 :

Ce contexte a été utilisé dans le(s) jugement(s) (2) :

Le fait 1 à l'état NIE
->

Le standard 5
->

Désirez-vous connaître les contextes d'application
de ce standard 5 ?(y/n)y

CONTEXTES DU STANDARD 5 :

CONTEXTE NECESSAIRE 7 défini le 1 1 1980 :

Le fait 2 à l'état AFFIRME défini par la LOI
->

Il n'existe pas ou plus de contexte d'application
du standard 5 .

Suite de la liste des contextes du standard 1 :

CONTEXTE SPECIFIQUE 1 associe au contexte 1 :

Ce contexte a été utilisé dans le(s) jugement(s) (1) :

Le fait 1 à l'état NIE
->

Il n'est plus possible de trouver d'autres contextes applicables.

Désirez-vous tester une autre combinaison de faits ?(y/n)n

Désirez-vous

1. Consulter les contextes d'un standard.
2. Connaître la meilleure argumentation à défendre pour une polarisation donnée.
3. Connaître la meilleure argumentation à défendre pour des faits donnés (et une polarisation donnée).
4. Mettre à jour les contextes d'un standard.
9. Retour au menu précédent.

Votre choix : 4

1. Ajout d'un contexte.
2. Suppression d'un contexte.
3. Modification d'un contexte.
9. Retour au menu précédent.

Votre choix : 1

A quel standard désirez-vous ajouter un contexte ?
Introduisez le numéro de ce standard : 5

:

Désirez-vous ajouter un

1. contexte nécessaire ?
2. contexte spécifique ?

Votre choix : 1

Introduisez les faits à ajouter
sous la forme ((<n°fait> <AFFIRME ou NIE> <origine>) etc.) :
((1 NIE LOI) (2 AFFIRME LOI))
Introduisez les standards à ajouter
sous la forme ((<n°standard> <origine>) etc.) :
()

CONTEXTE 10 AJOUTE
->

1. Ajout d'un contexte.
2. Suppression d'un contexte.
3. Modification d'un contexte.
9. Retour au menu précédent.

Votre choix : 1

A quel standard désirez-vous ajouter un contexte ?
Introduisez le numéro de ce standard : 5

Désirez-vous ajouter un

1. contexte nécessaire ?

2. contexte spécifique ?

Votre choix : 2

A quel contexte du standard 5 correspond le nouveau
contexte spécifique ? .

Introduisez les faits appartenant au contexte à ajouter
sous la forme ((<n°fait> <AFFIRME ou NIE>) etc.) :
((3 NIE))

Introduisez les standards appartenant au contexte à ajouter
sous la forme ((<n°standard> etc.) :

((11

Dans quel(s) jugement(s) est intervenu ce contexte ?
(sous la forme ((<n°jugement> etc.)) :

((1 2)

CONTEXTE 7 AJOUTE

->

1. Ajout d'un contexte.
2. Suppression d'un contexte.
3. Modification d'un contexte.
9. Retour au menu précédent.

Votre choix : 2

De quel standard désirez-vous supprimer un contexte ?
Introduisez le numéro de ce standard : 5

Désirez-vous supprimer un

1. Contexte nécessaire.
2. Contexte spécifique.

Votre choix : 1

Quel contexte nécessaire désirez-vous supprimer ?
Numéro de ce contexte : 10

Il y a au moins un contexte spécifique lié à ce contexte.
Il devra être supprimé.
->

CONTEXTE 10 SUPPRIME
->

1. Ajout d'un contexte.
2. Suppression d'un contexte.
3. Modification d'un contexte.
9. Retour au menu précédent.

Votre choix : 3

De quel standard désirez-vous modifier un contexte ?
Introduisez le numéro de ce standard : 1

Désirez-vous modifier un

1. Contexte nécessaire.
2. Contexte spécifique.

Votre choix : 2

Quel contexte spécifique désirez-vous modifier?
Numéro de ce contexte : 1

Désirez-vous mettre à jour les faits de ce contexte?(y/n)y

Désirez-vous

1. Ajouter des faits.
2. Supprimer des faits.
9. Ne pas modifier les faits.

Votre choix : 9

Désirez-vous mettre à jour les standards de ce contexte?(y/n)y

Desirez-vous

1. Ajouter des standards.
2. Supprimer des standards.
9. Ne pas modifier les standards.

Votre choix : 1

Introduisez les standards à ajouter
sous la forme (<n°standard>) :

(5)

Standard 5 ajouté.

Désirez-vous mettre à jour les liens entre ce contexte
et des jugements?(y/n)y

Desirez-vous

1. Ajouter des jugements respectant ce contexte.
2. Supprimer des jugements respectant ce contexte.
9. Ne pas modifier la jurisprudence.

Votre choix : 2

:

Introduisez les liens à supprimer avec les jugements
sous la forme : ((n°jugement) etc.)
(2)

Erreur d'introduction

CONTEXTE 1 MODIFIE

->

1. Ajout d'un contexte.
2. Suppression d'un contexte.
3. Modification d'un contexte.
9. Retour au menu précédent.

Votre choix : 9

Désirez-vous

1. Consulter les contextes d'un standard.
2. Connaître la meilleure argumentation à défendre pour une polarisation donnée.
3. Connaître la meilleure argumentation à défendre pour des faits donnés (et une polarisation donnée).
4. Mettre à jour les contextes d'un standard.
9. Retour au menu précédent.

Votre choix : 9

& QT.

A>

ANNEXE 2.1

RAPPORT DE STAGE CHEZ B. I. M. . EVERBERG.

1. Utilisation de BIM-Prolog.

BIM-Prolog est un langage de type PROLOG conçu et développé chez B. I. M. Il peut être utilisé sur VAX, système VMS ou UNIX.

Ses caractéristiques sont les suivantes :

- Définition de procédures de façon statique (non modifiables), ou dynamique (modifiables),
- Possibilité d'utiliser les commandes de l'O.S. à l'intérieur d'une session BIM-Prolog (ex : éditeur),
- Les variables doivent toutes débiter par le signe "_",
- Nombreux prédicats "built-in":
 - > Entrées/Sorties
 - > Contrôle
 - > Manipulation de règles
 - > Comparaisons
 - > Opérateurs.

J'ai pu tenter de réaliser un debugger permettant de suivre à l'écran les unifications de règles et instanciations de variables tout au long d'une exécution Prolog. Ce programme réalisé en BIM-Prolog n'a pu être achevé, car la ligne reliant le terminal de B. I. M. à la K. U. L. où se trouve le VAX était défectueuse. Une copie du texte du programme dans sa dernière version se trouve en Annexe 2.2.

2. Interface BIM-Prolog - Bases de données relationnelles.

Une possibilité très attrayante de BIM-Prolog est son interface avec le SGBD relationnel INGRES (une interface avec UNIFY est en cours de mise au point).

L'approche adoptée par BIM utilise une base de données conventionnelle qui résout le stockage des faits et la recherche, et l'intègre comme "back-end" à la partie déductive (Prolog). Ce qui signifie que les "queries" Prolog correspondant à des relations incluses dans la BD sont traduites en appels BD par l'interface.

Deux approches possibles sont l'approche interprétée et l'approche compilée. La première revient à évaluer les appels BD à l'exécution, en backtracking sur les solutions multiples éventuelles, tandis que la seconde suppose les procédures non récursives et applique les déclarations de procédures à la compilation, en propageant les valeurs des arguments et en instanciant les paramètres, jusqu'à obtenir une conjonction d'appels BD, qui sont alors passés au SGBD pour y être évalués.

La solution adoptée dans BIM-Prolog est intermédiaire entre ces deux approches et est appelée "évaluation partielle". Celle-ci consiste à évaluer les appels BD pour les procédures non-récursives, et à laisser pour l'exécution l'évaluation des procédures récursives. Le programme transformé par le méta-interpréteur Prolog d'évaluation partielle est alors une conjonction de prédicats built-in, d'appels à des règles récursives, et d'appels BD.

Pour l'utilisateur, la BD est aussi invisible que possible. Il lui suffit de déclarer les noms de relations se trouvant dans les BD externes comme des faits Prolog. Lors d'un query Prolog tel que :

```
?- address(Raf, _).
```

où address est le nom d'une relation appartenant à une BD externe déclarée et ouverte en lecture, l'interface transforme ce query en un appel BD :

```
retrieve ["address['Raf', '_']"<CR><LF>
```

Le SGBD prend en charge cet appel et renvoie à l'interface la reply :

"retrieve [[tuple], [pointers], return] <CR><LF>

où [tuple] = 1er élément de réponse

[pointers] = pointeurs vers les éléments de réponse suivants, utilisés par backtracking,

return est un code de retour signalant comment la request a été traitée.

Un exemple d'utilisation de cette interface se trouve en Annexe 2.3.

3. Programmes en BIM-Prolog.

J'ai pu voir exécuter trois programmes en BIM-Prolog :

1. Programme traitant les liens de parenté de la famille royale belge.
2. Programme d'acquisition de connaissances sur base de phrases réputées sémantiquement correctes.
3. Système expert évaluant pour un individu dont les caractéristiques sont données, l'attribution ou non d'une bourse d'études et son montant.

Une illustration des textes ou de l'exécution de ces programmes se trouve en annexes 2.4, 2.5 et 2.6.

4. Implémentation d'un langage de modélisation de concepts (CML) en Prolog.

Le but poursuivi est de présenter un outil de développement de logiciels. L'originalité de la méthode consiste à construire un formalisme interne de représentation des concepts en CProlog.

Les concepts de base de CML sont :

- CML = langage orienté-objet, objets apparaissant sous deux formes : "tokens" ou instances (occurrences particulières d'objets) et classes (concepts génériques).

- trois types d'objets de base :

Entités : objets, individus du monde décrit.

ex : personne, adresse,...

Assertions : Formules logiques et règles générales de représentation réglant l'attitude d'autres objets.

ex : règles de consistance.

Evénements : Activités résultant de changements dans l'univers du discours.

ex : enrôlement d'un étudiant à l'université.

- trois principes de base :

Classification : Grouper les objets en classes, les classes en méta-classes, etc., dont tous les éléments possèdent une propriété donnée.

Agrégation : Composition et décomposition de propriétés et d'entités (cfr. notion de record Pascal).

Généralisation/Hiérarchisation : Mécanisme d'abstraction permettant à des classes d'être des spécialisations d'autres classes (ex : "employé" de "personne") et d'hériter alors des propriétés accordées aux classes des niveaux supérieurs de la hiérarchie.

Catégorie : Relations entre objets.

ex : output(enrol, student, student) : l'output de
 l'activité "enrol" est l'entité "student" qui
est du type (classe) "student".

Ces catégories sont définies pour un objet, parmi une liste de catégories.

Deux exemples de définition CML se trouvent en annexes 2.7 et 2.8.

Une attention particulière des concepteurs sera d'offrir un environnement interactif agréable à l'utilisateur. Le langage de développement choisi pour supporter l'implémentation est Prolog, d'abord parce qu'il présente les mêmes richesses que CML, c'est-à-dire un langage orienté-objet, logique et fonctionnel, et ensuite parce que l'aspect "représentation des connaissances" et la vérification des assertions logiques étaient importants.

L'environnement interactif est orienté-menus.

L'opération fondamentale à réaliser dans le but de développement de logiciel est donc la traduction du langage CML en langage Prolog, impliquant des contrôles syntaxiques, de cohérence statique, du type "un objet ne peut être défini qu'une fois", et de complétude du système ("Tous les objets référencés sont définis").

Le système permet de modifier un objet, de décompiler sa traduction Prolog, d'avoir une vue graphique du système selon le modèle SADT, et de réaliser un prototype du système, permettant de le tester sur des données réelles tout en vérifiant la qualité des assertions et en testant les préconditions, postconditions et le nouvel état de la BD résultant de la simulation.

Ce projet n'étant pas encore complètement implémenté, les seules réalisations concrètes sont les traductions en Prolog (fichiers .PIF et .PEF) dont on trouve deux exemples en annexes 2.9 et 2.10.

Cette recherche s'inscrit dans un programme plus important développé dans le cadre du projet ESPRIT, le projet AIDA. AIDA est un outil d'aide à la conception d'un S.I. depuis l'analyse jusqu'à la programmation.

CML intervient à l'étape de modélisation et spécifications, en tant que langage de modélisation conceptuelle. L'environnement créé chez BIM permettra de tester la cohérence des spécifications.

La spécification détaillée et le prototypage du SI seront réalisés dans le langage de modélisation système SML et le langage de programmation fonctionnel-logique FLOP.

Les langages TAXIS et DBPL seront ceux du "software design" et de la production. Ce projet très ambitieux laisse au concepteur une liberté de choix lors de chaque transition d'étapes grâce à un environnement interactif (éditeurs, tests de validité, debuggers) approprié au langage utilisé.

Cette recherche étant toujours à l'état de projet, aucune réalisation concrète n'a pu être testée. Les promoteurs eux-mêmes sont hésitants sur la faisabilité de ce projet.

CML intervient également dans le projet LEARNCALC, lui aussi développé par la CEE. LEARNCALC sera un tableur électronique basé sur la logique.

Les avantages procurés par un langage logique dans la conception d'un spreadsheet sont, primo, une possibilité d'accéder directement à une BD Prolog, secundo, une utilisation plus flexible des relations entre cellules, tertio, la définition de cellules comme entités, textes, dessins, ou sous-tables.

Learncalc devrait avoir un mécanisme orienté-objet où les objets sont des entités, activités et assertions hiérarchisés et utilisant les principes de spécialisation et généralisation.

Le langage de modélisation CML et son environnement BIM-Prolog sont dès lors parfaitement appropriés, car directement exploitables par le spreadsheet qui serait implémenté en BIM-Prolog !

Le projet prévoit le développement d'une interface utilisateur agréable, et d'une méthode d'apprentissage par la machine de nouveaux concepts et manipulations, en utilisant les techniques d'analogie et d'exemples.

5. Recherche documentaire en Prolog.

Une base de données Prolog constituée d'un répertoire de livres et de mots-clés sert de support à un programme Prolog de consultation de cette base sur base de critères ou de combinaisons de critères tels que le nom de l'auteur, l'éditeur, l'année d'édition, etc.

Le programme existant au départ permettait de résoudre la recherche à partir d'une liste de mots ayant tous une signification pour la recherche.

Par exemple, [which, books, with, author, '=', marc] permet d'obtenir la liste des livres dont l'auteur est marc.

Le problème à résoudre ensuite était le suivant : permettre à l'utilisateur d'introduire directement une phrase en langage naturel au terminal sans passer par la liste de mots-clés.

Il a donc fallu ajouter une procédure transformant une phrase en langage naturel lue au terminal en une liste de mots-clés. Par exemple la phrase "Please, give me the list of the books who author is marc ?" aura le même effet que la liste citée dans l'exemple précédent.

6. Conclusion.

BIM-Prolog est un langage possédant des primitives élargies très utiles et réalisant des performances tout à fait appréciables en temps de réponse.

Ces quelques exercices de programmation m'ont fait découvrir les difficultés de ce langage, dont l'approche est très différente des langages de programmation traditionnels procéduraux, mais également ses possibilités, allant de l'implémentation de systèmes experts, ou de programmes d'acquisition de connaissances ou de recherche documentaire, jusqu'au spreadsheet électronique.

Prolog augmenté d'une interface BD relationnelle devient un outil encore plus performant, en ajoutant aux possibilités des SGBD une dimension logique déductive. Prolog apparaît comme un outil approprié au développement de logiciels résolvant un S.I. spécifié en CML, base de l'implémentation finale.

Ce séjour m'a donc permis de tester des réalisations concrètes rédigées en Prolog, dans des domaines très divers.

ANNEXE 2.2

DEBUGGER INTERACTIF EN PROLOG.

```
:- alldynamic.

debug(_x):-
    debugdec(_x, 0).

debugdec((_x, _y), _niv):-
    !,
    spaces(_niv), write(_x), nl,
    (clause(_x, _ssbuts, _nv),
     resoudre(_x, _ssbuts, _niv, _nr);
     (builtin(_x), call(_x), nl, resoudre(_x, 'true', _niv, 0);
      spaces(_niv), write('*'), spaces(1), write(_x),
      write('ECHEC'), nl, fail)),
    ecriret(_niv),
    debugdec(_y, _niv).

debugdec(_x, _niv):-
    spaces(_niv), write(_x), nl,
    (clause(_x, _ssbuts, _nr),
     resoudre(_x, _ssbuts, _niv, _nr);
     (builtin(_x), call(_x), nl, resoudre(_x, 'true', _niv, 0);
      spaces(_niv), write('*'), spaces(1), write(_x),
      write('ECHEC'), nl, fail)).

ecriret(_niv):-
    spaces(_niv),
    write('&'),
    nl.

resoudre(_x, 'true', _niv, _nr):-
    !, spaces(_niv), write(_nr), spaces(1), write(_x),
    write('SUCCES'), nl.

resoudre(_x, _ssbuts, _niv, _nr):-
    assert((_x: -, info(_x, _ssbuts, _niv, _nr, _rep),
            tracer(_rep, _ssbuts, _niv)), 1),
    (call(_x), !, spaces(_niv), write(_nr), spaces(1),
     write(_x), write('SUCCES'), nl,
```

```

    retract((_x:-_), 1);
    spaces(_niv), write(_nr), spaces(1), write(_x),
    write('ECHEC'), n),
    retract((_x:-_), 1), fail).

info(_x, _ssbuts, _niv, _nr, _rep):-
    spaces(_niv),
    write(_nr),
    spaces(1), write(_x),
    write('Trace?(y./n.)'),
    read(_rep).

tracer('y', _ssbuts, _niv):-
    !, incrniv(_niv, _new),
    debugdec(_ssbuts, _new).

tracer(_x, _ssbuts, _niv):-
    call(_ssbuts).

incrniv(_old, -new):-
    -new is _old + 1.

```


ANNEXE 1

MODE D'UTILISATION

DU PROTOTYPE D'AIDE A L'ARGUMENTATION JUDICIAIRE.

Les pages suivantes donnent un exemple d'utilisation complète du prototype. Les instructions nécessaires à son déroulement y sont soulignées, à l'inverse des réponses du programme. Lorsqu'apparaît à l'écran une flèche ->, cela signifie qu'il faut introduire n'importe quel caractère pour que le déroulement se poursuive.

La session commence à partir du système d'exploitation. Il suffit d'une disquette, dans laquelle se trouvent les fichiers Micro-Prolog, plus les fichiers du prototype.

Le fichier BASE.LOG doit contenir au moins des faits, des standards et des jugements, sous forme de faits Prolog.

Le fichier BASE.LOG au début de la session est le suivant :

```
((derndontnee 9))
((derndontspec 6))
((fait 1 "TEXTE DU FAIT 1" +))
((fait 2 "TEXTE DU FAIT 2" -))
((fait 3 "TEXTE DU FAIT 3" +))
((fait 4 "TEXTE DU FAIT 4" -))
((standard 1 "TEXTE DU STANDARD 1" - AFFIRME 0))
((standard 2 "TEXTE DU STANDARD 2" + AFFIRME 0))
((standard 3 "TEXTE DU STANDARD 3" - AFFIRME 0))
((standard 4 "TEXTE DU STANDARD 4" + AFFIRME 0))
((standard 5 "TEXTE DU STANDARD 5" - AFFIRME 0))
((contnee 1 1 1 1 1980))
((contnee 2 1 1 1 1980))
((contnee 3 2 1 1 1980))
((contnee 4 3 1 1 1980))
((contnee 5 3 1 1 1980))
((contnee 6 3 1 1 1980))
((contnee 7 5 1 1 1980))
((contnee 8 4 0 0 0))
```

((valide 1 1))
 ((valide 1 2))
 ((valide 2 1))
 ((valide 2 3))
 ((valide 3 2))
 ((valide 3 3))
 ((valide 4 1))
 ((valide 4 2))
 ((valide 4 3))
 ((valide 5 1))
 ((valide 5 2))
 ((valide 5 3))
 ((valide 6 2))
 ((valide 6 3))
 ((condnecfait 0 0 0 0))
 ((condnecfait 1 2 AFFIRME LOI))
 ((condnecfait 1 3 NIE LOI))
 ((condnecfait 2 1 NIE LOI))
 ((condnecfait 2 2 AFFIRME CASSATION))
 ((condnecfait 3 1 AFFIRME CASSATION))
 ((condnecfait 4 3 NIE LOI))
 ((condnecfait 5 1 NIE LOI))
 ((condnecfait 6 2 AFFIRME CASSATION))
 ((condnecfait 6 1 NIE CASSATION))
 ((condnecfait 7 2 AFFIRME LOI))
 ((condnecfait 0 0 0))
 ((condnecfait 1 3 LOI))
 ((Jugement 1 860304 NAMUR "CASSATION SECONDE RENVOI" LIBERATION NON))
 ((Jugement 2 830201 NAMUR "PREMIER DEGRE" LIBERATION OUI))
 ((Jugement 3 860235 LIEGE "CASSATION" MAINTIEN NON))
 ((contspec 0 0 0))
 ((contspec 1 1 1))
 ((contspec 2 1 1))
 ((contspec 3 2 3))
 ((contspec 4 3 5))
 ((contspec 5 4 0))
 ((contspec 6 4 0))
 ((dans 0 0))
 ((dans 1 1))
 ((dans 2 2))
 ((dans 3 1))
 ((dans 4 1))
 ((dans 5 2))
 ((dans 6 3))
 ((condspecfait 0 0 0))
 ((condspecfait 1 1 NIE))
 ((condspecfait 2 1 NIE))
 ((condspecfait 3 2 NIE))
 ((condspecfait 3 3 AFFIRME))
 ((condspecfait 4 3 NIE))
 ((condspecfait 5 1 AFFIRME))
 ((condspecfait 6 3 AFFIRME))
 ((condspecstand 0 0))
 ((condspecstand 2 5))
 ((condspecstand 6 2))

ANNEXE 2.3.1

Sep 23 12:15 1985 dba.pro Page 1

```
1  :- extern(address, 2, testdb).
2  :- extern(steden, 3, testdb).
3  :- extern(hc, 2, testdb).
4
5  o :- opendb(testdb).
6  c :- closedb.
7  cr :- createrelation(hc(voor(c20,key),achter(c20,nokey)), isam, testdb).
8  dr :- deleterelation(hc, testdb).
9  r(_x) :- retrieve(_x, testdb), writeq(_x), nl.
10 i(_x) :- insert(_x, testdb) .
11 d(_x) :- delete(_x, testdb) .
12
```

```

1 Script started on Mon Sep 23 11:53:58 1985
2 sunbim% BIMprolog dba.pro
3 BIMprolog - release 1.05 9-9-1985
4 trying to compile file dba.pro
5 dba.pro consulted.
6
7 > ?- o .
8
9 > ?- r( steden(_x, _y, _z) ) .
10 steden(Dworp,Brabant,1512)
11 steden(Flawinne,Namen,5720)
12 steden(Vorst,Brabant,1190)
13 steden('St.Truiden',Limburg,3800)
14 steden(Kortrijk,'West-Vlaanderen',8500)
15 steden(Bergen,Henegouwen,7000)
16 steden(Antwerpen,Antwerpen,2000)
17 steden(Luik,Luik,4000)
18 steden(Hasselt,Limburg,3500)
19 steden(Diepenbeek,Limburg,3610)
20 steden(Brussel,Brabant,1000)
21 steden(Zolder,Limburg,3540)
22 steden(Leuven,Brabant,3000)
23 steden(Brasschaat,Antwerpen,2130)
24 steden(Gent,'Oost-Vlaanderen',9000)
25 steden(Genk,Limburg,3600)
26 steden(Namen,Namen,5000)
27 steden(Heverlee,Brabant,3030)
28
29 > ?- steden(_x, _y, _z), write(steden(_x, _y, _z),)p nl .
30 steden(Dworp,Brabant,1512)
31 steden(Flawinne,Namen,5720)
32 steden(Vorst,Brabant,1190)
33 steden('St.Truiden',Limburg,3800)
34 steden(Kortrijk,'West-Vlaanderen',8500)
35 steden(Bergen,Henegouwen,7000)
36 steden(Antwerpen,Antwerpen,2000)
37 steden(Luik,Luik,4000)
38 steden(Hasselt,Limburg,3500)
39 steden(Diepenbeek,Limburg,3610)
40 steden(Brussel,Brabant,1000)
41 steden(Zolder,Limburg,3540)
42 steden(Leuven,Brabant,3000)
43 steden(Brasschaat,Antwerpen,2130)
44 steden(Gent,'Oost-Vlaanderen',9000)
45 steden(Genk,Limburg,3600)
46 steden(Namen,Namen,5000)
47 steden(Heverlee,Brabant,3030)
48
49 > ?- cr .
50
51 > ?- l( hc( jef, jules) ) .
52
53 > ?- r( hc(_x, _y) ) .
54 hc(jef,jules)
55
56 > ?- hc(_x, _y), write(_x), spaces(2), write(_y), nl .

```

```

57 jef jules
58
59
60
61
62
63 BIMprolog - release 1.05 9-9-1985
64 trying to compile file /bim/pprolog/non/startup.pro
65 dba.pro consulted.
66
67 > ?- o .
68
69 > ?- r( hc(_x, _y) ) .
70 hc(jef,jules)
71
72 > ?- assert( hc(an, marie) ) .
73
74 > ?- r( hc(_x, _y) ) .
75 hc(jef,jules)
76 hc(an,marie)
77
78 > ?- l( hc( bart, janine) ) .
79
80 > ?- r( hc(_x, _y) ) .
81 hc(jef,jules)
82 hc(bart,janine)
83 hc(an,marie)
84
85 > ?- d( hc( jef, marie) ) .
86
87 > ?- delete( hc( jef, _x) ), write(_x), nl .
88 *** RUNTIME 220 *** Illegal call : unknown procedure delete/1.
89
90 > ?- delete( hc( jef, _x) , t66bdb) , write(_x), nl .
91
92 *** SYN_ERR 9 *** '.' expected at end of clause
93
94 > ?- delete( hc( jef, _x) , testdb) , write(_x), nl .
95 marie
96 jules
97
98 > ?- d( hc(_x, _y) ), write( w(_x, _y) ), nl .
99 w(bart,janine)
100 w(an,marie)
101
102 > ?- dr .
103
104 > ?- c .
105
106 > ?- steden(_x, _y, _z), write(a (_x, _y, _z) ), nl .
107

```



```

1 Script started on Mon Sep 23 12:21:11 1985
2 prolog diana\ vorst.pro
3 BIMprolog - release 1.05 9-9-1985
4 vorst.pro consulted.
5
6
7
8 this program gives info about the Belgian royal family
9
10 written by Herman Crauwels ??-??-1985
11
12 BIM
13 Kwikstraat 4
14 3078 Everberg
15 BELGIUM
16
17
18 help could help you
19
20 QQ help .
21
22 Type vragen :
23
24 [wie is de] <relatie> van <persoon> .
25 is <persoon> [de] <relatie> van <persoon> .
26 [geef de] <relatie> tussen <persoon> en <persoon> .
27 [de] relaties van <persoon> .
28 aantal <relatie> van <persoon> .
29 gezin van <persoon> .
30 gezin van <persoon> vertikaal .
31 keten van <persoon> .
32 help .
33 einde .
34
35 <relatie> :
36
37 echtgenoot vader grootvader oom voorvader
38 echtgenote moeder grootmoeder tante
39 broer zoon kleinzoon neef nakomeling
40 zuster dochter kleindochter nicht
41
42 of de meervoudsvormen
43
44
45 QQ vader van BO\177oudewijn I
46
47 QQ vader van Boudewijn I
48 Leopold III
49
50 QQ relaties tussen Albert I en Filipske
51
52 QQ relatie tussen Albert I en Filipske
53 voorvader
54
55 QQ relaties van Albert I
56 vader van Leopold III

```

```

57 vader van Karel
58 vader van Marie-Jose
59 echtgenoot van Elisabeth
60 zoon van Filips
61 zoon van Maria
62 grootvader van Josephina-Charlotta
63 grootvader van Boudewijn I
64 grootvader van Albert
65 kleinzoon van Louisa-Maria
66 kleinzoon van Leopold I
67 broer van Boudewijn
68 broer van Henrica
69 broer van Josephinake
70 broer van Josephina
71 neef van Lodewijk-Filips
72 neef van Leopold II
73 neef van Maxmilliaan
74 neef van Charlotta
75 neef van Maria-Henrica
76 neef van Louisa
77 neef van Leopold
78 neef van Stefanie
79 neef van Clementina
80 nakomeling van Filips
81 nakomeling van Leopold I
82 voorvader van Leopold III
83 voorvader van Karel
84 voorvader van Marie-Jose
85 voorvader van Josephina-Charlotta
86 voorvader van Boudewijn I
87 voorvader van Albert
88 voorvader van Filipske
89 voorvader van Astridje
90
91 QQ einde
92 EINDE
93
94 QQ einde
95 EINDE
96
97 > dDana\ ^D
98 script done on Mon Sep 23 12:24:38 1985

```

```

1 [article of Alain Grumbach Knowledge Aquisition in Prolog: 20-1-1985]
2
3 :- dynamic link_sem/2 .
4
5 class_sem(lenina,human) .
6 class_sem(eats,eat) .
7 class_sem(drinks,drink) .
8 class_sem(cake,food) .
9 class_sem(coffee,beverage) .
10 class_sem(tea,beverage) .
11 class_sem(henri,human) .
12
13 show_class_sem :- listing(class_sem/2) .
14
15 link_sem(human,eat) .
16 link_sem(eat,food) .
17 link_sem(human,drink) .
18
19 show_link_sem :- listing(link_sem/2) .
20
21 test :- nl , lees(_p) , nl ,
22         schrijflijst(_p) , write(' is semantically '),
23
24         (sem(_p,_a) , schrijfanswer(_a) , ! ;
25
26         write(wrong) , nl) .
27
28
29 make_true :- nl , lees(_p) , nl ,
30             istrue(_p) .
31
32
33 schrijfanswer(yes) :- write(correct) , nl .
34 schrijfanswer(no) :- write(wrong) , nl .
35
36 schrijflijst([]) .
37 schrijflijst([_x|_y]) :- write(_x) , write(' '), schrijflijst(_y) .
38
39
40 sem(_p,_a):-sem_class(_p,_np) ,
41             sem_verify(_np,_a) , ! .
42
43
44 sem_class([],[]) .
45 sem_class([_x|_z],[_nx|_nz]) :- class_sem(_x,_nx) , sem_class(_z,_nz) .
46
47
48 sem_verify([_x],yes) :- ! .
49 sem_verify([_a,_b|_c],_answer) :- link_sem(_a,_b) , ! ,
50                                 sem_verify([_b|_c],_answer) .
51
52 sem_verify(_,no) .
53
54 istrue(_p) :- assert((link_sem(_x,_y) :- assert(link_sem(_x,_y)))) ,
55             sem(_p,yes) ,
56

```

```

57             retract((link_sem(_x,_y) :- assert(link_sem(_x,_y))))
58
59
60
61 lees(_l) :- leesl([],_m) ,
62             transform(_m,_ll) ,
63
64             reversel(_ll,[],_l) .
65
66
67 leesl(_in,_uit) :- readc(_x) , not(ascii(_x,10)) ,
68                     ! ,
69
70                     leesl([_x|_in],_uit) .
71
72 leesl(_in,_in) .
73
74 transform(_m,[_me|_nrest]) :- append(_e,[' '|_rest],_m) ,
75                                 ! ,
76
77                                 reverse(_e,[],_ne) ,
78
79                                 atomtolist(_me,_ne) ,
80
81                                 transform(_rest,_nrest) .
82
83 transform(_l,_j) :- reverse(_l,[],_nl) , atomtolist(_j,_nl) .
84
85
86 append([],_x,_x) .
87 append([_x|_l],_l2,[_x|_l3]) :- append(_l,_l2,_l3) .
88
89 reversel([_x|_l],_in,_uit) :- ! , reversel(_l,[_x|_in],_uit) .
90 reversel(_x,_l,[_x|_l]) .
91 reverse([],_l,_l) .
92 reverse([_x|_l],_in,_uit) :- reverse(_l,[_x|_in],_uit) .
93
94
95 lis :- write('class_sem'),nl,
96         while(class_sem(_X,_Y),writenl([_X,_Y])),nl,
97         write('link sem'),nl,
98         while(link_sem(_X,_Y),writenl([_X,_Y])),nl.
99
100 while(_X,_Y) :- _X, _Y, fail.
101 while(_X,_Y).
102
103 writenl(_X) :- write(_X),nl.
104
105 loop :- nl , nl , write('make your choice: ') ,
106
107         write('          ') , write('test          t') , nl ,
108
109         write('          ') , write('make true      m') , nl ,
110
111         write('          ') , write('show db        s') , nl ,
112
113         write('          ') , write('exit          e') , nl ,

```


ANNEXE 2.5.2

Sep 23 12:34 1985 know.pro Page 3

```
113
114     readc(_x) , readc(_y) , action(_x) , fail .
115 loop :- loop .
116
117 action(t) :- ! , nl , write('enter sentence to test') , nl , test .
118 action(m) :- ! , nl , write('enter sentence to assert ') , nl , make_true .
119 action(s) :- ! , nl , write('current data base :') , nl , lis .
120 action(e) :- nl , write('bye bye') , nl , exit .
121
122 ?- loop .
```

```

1 diana% ../non/BIMprolog yapesp.pro studbrs.pro
2 BIMprolog - release 1.05 9-9-1985
3 yapesp.pro consulted.
4
5
6
7
8           Expert System Shell   Crickets
9
10                  Bruno Krekels
11      K.U.Leuven, Dept. of Computer Science
12      Celestijnelaan 200A, 3030 Leuven, Belgium
13
14
15
16
17
18
19 studbrs.pro consulted.
20
21 > ?- bewijs( beurs(ikke, _x) ) .
22
23
24 Over welk akademiejaar gaat het   ?
25 antwoord is
26 @ 1986 .
27
28
29 Is ikke gehuwd   ?
30 antwoord is
31 @ nee .
32
33
34 Is ikke fiscaal ten laste van zijn ouders voor 31 / 12 / 1986   ?
35 antwoord is
36 @ ja .
37
38
39 Is ikke een regelmatige student   ?
40 antwoord is
41 @ ja .
42
43
44 Is er na 1984 een verandering in de toestand van ikke opgetreden   ?
45 antwoord is
46 @ nee .
47
48
49 Is er een herziening van de belastingaanslag gebeurd bij
50           ouders(ikke) in 1984   ?
51 antwoord is
52 @ nee .
53
54 Gebruik het aanslagbiljet van 1984 om de inkomsten van
55           ouders(ikke) te bepalen
56

```

```

57
58 Hoeveel bedraagt in 1984 het gezamenlijk belastbaar inkomen van
59           ouders(ikke)   ?
60 antwoord is
61 @ 200000 .
62
63
64 Hoeveel bedraagt in 1984 het afzonderlijk belastbaar inkomen van
65           ouders(ikke)   ?
66 antwoord is
67 @ 0 .
68
69
70 Hoeveel bedraagt in 1984 het belastbaar kadastraal inkomen van
71           ouders(ikke)   ?
72 antwoord is
73 @ 0 .
74
75
76 Hoeveel bedraagt in 1984 het gezamenlijk belastbaar inkomen van
77           ikke   ?
78 antwoord is
79 @ 0 .
80
81
82 Hoeveel bedraagt in 1984 het afzonderlijk belastbaar inkomen van
83           ikke   ?
84 antwoord is
85 @ 0 .
86
87
88 Hoeveel bedraagt in 1984 het belastbaar kadastraal inkomen van
89           ikke   ?
90 antwoord is
91 @ 0 .
92
93
94 Hoeveel personen heeft ouders(ikke) ten laste   ?
95 antwoord is
96 @ 4 .
97
98
99 Hoeveel personen ten laste van ouders(ikke) volgen hogere studies
100           in 1986   ?
101 antwoord is
102 @ 2 .
103
104
105 Neemt ikke logies in de universiteitsstad voor 31 / 12 / 1986   ?
106 antwoord is
107 @ ja .
108
109 Oplossing gevonden
110           de studiebeurs voor ikke bedraagt 74300 fr
111
112

```


113 Stop(s), Waarom(w) of Backtracken(b) ?
114 @ w
115 .
116
117
118 1 de studiebeurs voor ikke bedraagt 74300 fr
119
120 Geef een nummer om meer uitleg :
121 @ 1 .
122
123
124 de studiebeurs voor ikke bedraagt 74300 fr want
125 1 ikke is afhankelijk van ouders(ikke) voor 31 / 12 / 1986
126 volgens art(1) en
127 2 ikke is een regelmatige student en
128 3 ouders(ikke) moet als totaalinkomen 200000 fr rekenen op basis
129 van 1984 en
130 4 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2)
131 5 ikke krijgt 1 van de maximale beurs, dus 74300 fr en
132
133 Geef een nummer om meer uitleg :
134 @ 4 .
135
136
137 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2) want
138 1 ouders(ikke) heeft 4 personen ten laste en
139 2 1986 is het beschouwde academiejaar en
140 3 in 1986 volgen 2 personen hogere studies ten laste van
141 ouders(ikke) en
142 4 5 is 4 + 2 - 1
143
144 Geef een nummer om meer uitleg :
145 @ 4 .
146
147
148 5 is 4 + 2 - 1
149 is een systeem oproep
150
151
152
153 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2) want
154 1 ouders(ikke) heeft 4 personen ten laste en
155 2 1986 is het beschouwde academiejaar en
156 3 in 1986 volgen 2 personen hogere studies ten laste van
157 ouders(ikke) en
158 4 5 is 4 + 2 - 1
159
160 Geef een nummer om meer uitleg :
161 @ 0 .
162
163 de studiebeurs voor ikke bedraagt 74300 fr want
164 1 ikke is afhankelijk van ouders(ikke) voor 31 / 12 / 1986
165 volgens art(1) en
166 2 ikke is een regelmatige student en
167 3 ouders(ikke) moet als totaalinkomen 200000 fr rekenen op basis
168 van 1984 en

169 4 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2) e
170 5 ikke krijgt 1 van de maximale beurs, dus 74300 fr
171
172 Geef een nummer om meer uitleg :
173 @ 5 .
174
175
176 ikke krijgt 1 van de maximale beurs, dus 74300 fr want
177 1 met 200000 fr inkomsten en 5 personen ten laste heeft ikke recht
178 op 1 deel van de maximum toelage volgens art(10,1) en
179 2 1 van de maximumtoelage voor ikke geeft 74300 fr volgens art(13)
180
181 Geef een nummer om meer uitleg :
182 @ 2 .
183
184
185 1 van de maximumtoelage voor ikke geeft 74300 fr volgens art(13) want
186 1 de maximum beurs voor ikke bedraagt 74300 volgens art(10,2) en
187 2 74300 is 74300 * 1 en
188 3 het afgerond bedrag van 74300 is 74300 fr volgens art(13)
189
190 Geef een nummer om meer uitleg :
191 @ 0- .
192
193 ikke krijgt 1 van de maximale beurs, dus 74300 fr want
194 1 met 200000 fr inkomsten en 5 personen ten laste heeft ikke recht
195 op 1 deel van de maximum toelage volgens art(10,1) en
196 2 1 van de maximumtoelage voor ikke geeft 74300 fr volgens art(13)
197
198 Geef een nummer om meer uitleg :
199 @ 0 .
200
201 de studiebeurs voor ikke bedraagt 74300 fr want
202 1 ikke is afhankelijk van ouders(ikke) voor 31 / 12 / 1986
203 volgens art(1) en
204 2 ikke is een regelmatige student en
205 3 ouders(ikke) moet als totaalinkomen 200000 fr rekenen op basis
206 van 1984 en
207 4 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2)
208 5 ikke krijgt 1 van de maximale beurs, dus 74300 fr
209
210 Geef een nummer om meer uitleg :
211 @ 0 .
212
213
214 1 de studiebeurs voor ikke bedraagt 74300 fr
215
216 Geef een nummer om meer uitleg :
217 @ 0 .
218
219
220 Einde_uitleg ...
221
222 Stop(s), Waarom(w) of Backtracken(b) ?
223 @ b .
224

ANNEXE 2.6.2

```

225 Een nieuwe oplossing wordt gezocht...
226
227
228 Geen oplossing (meer) voor
229 de studiebeurs voor ikke bedraagt --- fr
230
231
232 Waarom_niet (ja/nee) ?
233 @ ja .
234
235
236 1 : de studiebeurs voor ikke bedraagt --- fr          indien
237 ikke is afhankelijk van Sponsor voor Datum volgens Art en
238 ikke is een regelmatige student en
239 Sponsor moet als totaalinkomen Inkomen fr rekenen op basis van
240 Inkomenjaar en
241 Sponsor mag Aantal personen ten laste rekenen volgens Art en
242 ikke krijgt Coefficient van de maximale beurs, dus --- fr
243
244 Geen regels meer voor
245 de studiebeurs voor ikke bedraagt --- fr
246
247
248 Geef nummer om meer waarom-niet uitleg :
249 @ 1 .
250
251
252 De 1_de regel :
253
254 Er is een oplossing !!
255
256 de studiebeurs voor ikke bedraagt 74300 fr          indien
257 ikke is afhankelijk van ouders(ikke) voor 31 / 12 / 1986
258 volgens art(1) ... is waar
259 ikke is een regelmatige student ... is waar
260 ouders(ikke) moet als totaalinkomen 200000 fr rekenen op basis
261 van 1984 ... is waar
262 ouders(ikke) mag 5 personen ten laste rekenen volgens art(15,2)
263 ikke krijgt 1 van de maximale beurs, dus 74300 fr ... is waar
264
265 Waarom(w), Backtracken(b) of "ok" ? :
266 @ ok .
267
268
269 1 : de studiebeurs voor ikke bedraagt --- fr          indien
270 ikke is afhankelijk van Sponsor voor Datum volgens Art en
271 ikke is een regelmatige student en
272 Sponsor moet als totaalinkomen Inkomen fr rekenen op basis van
273 Inkomenjaar en
274 Sponsor mag Aantal personen ten laste rekenen volgens Art en
275 ikke krijgt Coefficient van de maximale beurs, dus --- fr
276
277 Geen regels meer voor
278 de studiebeurs voor ikke bedraagt --- fr
279
280

```

```

281 Geef nummer om meer waarom-niet uitleg :
282 @ 0 .
283
284 Einde waarom_niet uitleg.
285
286
287 > ?_bewijs( beurs( hij, _y ) ) .
288
289
290 Is hij gehuwd ?
291 antwoord is
292 @ nee .
293
294
295 Is hij fiscaal ten laste van zijn ouders voor 31 / 12 / 1986 ?
296 antwoord is
297 @ ja .
298
299
300 Is hij een regelmatige student ?
301 antwoord is
302 @ ja .
303
304
305 Is er na 1984 een verandering in de toestand van hij opgetreden ?
306 antwoord is
307 @ nee .
308
309
310 Is er een herziening van de belastingsaanslag gebeurd bij
311 ouders(hij) in 1984 ?
312 antwoord is n
313 @ ee .
314
315 Gebruik het aanslagbiljet van 1984 om de inkomsten van ouders(hij)
316 te bepalen
317
318 ... is waar
319 Hoeveel bedraagt in 1984 het gezamenlijk belastbaar inkomen van
320 ouders(hij) ?
321 antwoord is
322 @ 1000000 .
323
324
325 Hoeveel bedraagt in 1984 het afzonderlijk belastbaar inkomen van
326 ouders(hij) ?
327 antwoord is
328 @ 0 .
329
330
331 Hoeveel bedraagt in 1984 het belastbaar kadastraal inkomen van
332 ouders(hij) ?
333 antwoord is
334 @ 0 .
335
336

```


337 Hoeveel bedraagt in 1984 het gezamenlijk belastbaar inkomen van
 338 hij ?
 339 antwoord is
 340 @ 0 .
 341
 342
 343 Hoeveel bedraagt in 1984 het afzonderlijk belastbaar inkomen van
 344 hij ?
 345 antwoord is
 346 @ 0 .
 347
 348
 349 Hoeveel bedraagt in 1984 het belastbaar kadastraal inkomen van hij ?
 350 antwoord is
 351 @ 0 .
 352
 353
 354 Hoeveel personen heeft ouders(hij) ten laste ?
 355 antwoord is
 356 @ 1 .
 357
 358
 359 Hoeveel personen ten laste van ouders(hij) volgen hogere studies
 360 in 1986 ?
 361 antwoord is
 362 @ 1 .
 363
 364
 365 Geen oplossing (meer) voor
 366 de studiebeurs voor hij bedraagt --- fr
 367
 368
 369 Waarom_niet (ja/nee) ?
 370 @ ja .
 371
 372
 373 1 : de studiebeurs voor hij bedraagt --- fr indien
 374 hij is afhankelijk van Sponsor voor Datum volgens Art en
 375 hij is een regelmatige student en
 376 Sponsor moet als totaalinkomen Inkomen fr rekenen op basis van
 377 Inkomenjaar en
 378 Sponsor mag Aantal personen ten laste rekenen volgens Art en
 379 hij krijgt Coefficient van de maximale beurs, dus --- fr
 380
 381 Geen regels meer voor
 382 de studiebeurs voor hij bedraagt --- fr
 383
 384
 385 Geef nummer om meer waarom-niet uitleg :
 386 @ 1 .
 387
 388
 389 De 1_de regel :
 390
 391 de studiebeurs voor hij bedraagt --- fr indien
 392 hij is afhankelijk van ouders(hij) voor 31 / 12 / 1986 volgens

393 art(1) ... is waar
 394 hij is een regelmatige student ... is waar
 395 ouders(hij) moet als totaalinkomen 1000000 fr rekenen op basis
 396 van 1984 ... is waar
 397 ouders(hij) mag 1 personen ten laste rekenen volgens art(15,2)
 398 hij krijgt Coefficient van de maximale beurs, dus Som fr
 399 ... is NIET waar
 400
 401 Waarom-niet(wn) , Reden-Mislukking(rm) , Backtracken(b) of "ok" :
 402 @ rm .
 403
 404
 405
 406 1000000 <= 44100
 407 is NIET waar
 408
 409
 410 1000000 <= 241800
 411 is NIET waar
 412
 413
 414 1000000 < 441000
 415 is NIET waar
 416
 417
 418 Alle oproepen die geen passende oplossing hadden werden getoond
 419
 420
 421
 422 de studiebeurs voor hij bedraagt --- fr indien
 423 hij is afhankelijk van ouders(hij) voor 31 / 12 / 1986 volgens
 424 art(1) ... is waar
 425 hij is een regelmatige student ... is waar
 426 ouders(hij) moet als totaalinkomen 1000000 fr rekenen op basis
 427 van 1984 ... is waar
 428 ouders(hij) mag 1 personen ten laste rekenen volgens art(15,2)
 429 hij krijgt Coefficient van de maximale beurs, dus Som fr
 430 ... is NIET waar
 431
 432 Waarom-niet(wn) , Reden-Mislukking(rm) , Backtracken(b) of "ok" :
 433 @ ok .
 434
 435
 436 1 : de studiebeurs voor hij bedraagt --- fr indien
 437 hij is afhankelijk van Sponsor voor Datum volgens Art en
 438 hij is een regelmatige student en
 439 Sponsor moet als totaalinkomen Inkomen fr rekenen op basis van
 440 Inkomenjaar en
 441 Sponsor mag Aantal personen ten laste rekenen volgens Art en
 442 hij krijgt Coefficient van de maximale beurs, dus --- fr
 443
 444 Geen regels meer voor
 445 de studiebeurs voor hij bedraagt --- fr
 446
 447
 448 Geef nummer om meer waarom-niet uitleg :

ANNEXE 2.6.5

Sep 23 12:46 1985 proberen3 Page 9

```
449 @ 0 .  
450  
451 Einde waarom_niet uitleg.  
452  
453  
454 > ddana^ ^D  
455 script done on Mon Sep 23 12:45:31 1985
```


ANNEXE 2.7

Ex2.CML

```
def      admit_patient
in      activity_class
with
  input
    P:person.
  control
    w:ward;
    phys, consulting_phys : physician.
  output
    pt:patient.
  actcond
    a1:arrival(P).
  precond
    already_in? : not(inst(P,patient));
    room_left? : lt(of(cardinality,patient),patient_max).
  postcond
    admitted? : in_hospital(P).
  part
    chech_id : chech_id(P);
    temp : take_temp(P)
end#

def      admit_child_patient
in      activity_class
use     admit_patient
with
  input
    P : child.
  control
    n : nurse.
  output
    pt : child_patient.
  part
    find_nurse : find_nurse(n,w);
    admit : insert(P,child_patient)
end#

def      admit_surgical_patient
in      activity_class
use     admit_patient
with
  output
    pt : surgical_patient.
  actcond
    - a1 : and(arrival(P),surgery_needed(P)).
  part
    blood_typing : perform_blood_typing
end#

def      admit_surgical_child_patient
in      activity_class
use     admit_child_patient,admit_surgical_patient
with
  part
    obtain_permission : obtain_permission(P,of(guardian,P))
end#
```

ANNEXE 2.8.1

Ex3 c n l

```
def     entre
in      entity_class
isa     entity
with
    argument
        INF,SUP : number.
    necond
        n1 : number(THIS);
        n2 : le(THIS,SUP);
        n3 : le(INF,THIS)
endf
```

```
def     date
in      entity_class
isa     entity
with
    part
        year : entre(45,84);
        mounth : entre(1,12);
        day : entre(1,31)
endf
```

```
def     today
in      date
with
    part
        year : 84;
        mounth : october;
        day : 23
endf
```

```
def     person
in      entity_class
isa     entity
with
    part
        nom : string(20);
        prenom : string(20);
        adresse : adresse;
        tel : integer(9);
        naiss : date;
        age : @ (person_age)
endf
```

```
def     person_age
in      activity_class
with
    .control
        THIS : person.
    output
        AGE : integer(3).
    part
        a1 : is(AGE,@(moins(of(year,today),of(year,of(naiss,THIS))))))
endf
```

```
def     adresse
in      entity_class
isa     entity
with
    part
        rue : string(30);
```


EV3.CNL

(c)

```
no : integer(3);
code_postal : integer(4);
localite : string(20)

endf

def integer
in entity_class
with
argument
I : number.
precond
P : number(THIS).
neccond
n : lt(THIS,@(exp(10,I)))
endf

def string
in entity_class
with
argument
I : integer(3).
precond
P : atom(THIS).
neccond
n : le(@(strlen(THIS)),I)
endf

def number
in entity_class
with
neccond
n : number(THIS)
endf

def js
in person
with
part
nom : smith;
prenom : john;
adresse : adresse_js;
tel : 27595925;
naiss : naiss_js
endf

def naiss_js
in date
with
part
year : 60;
mounth : 6;
day : 8
endf

def adresse_js
in adresse
with
part
rue : du_moulin;
```

ANNEXE 2.8.3

Ex 2. CML

```
code_postal : 1000;  
localite : bruxelles  
endf
```



```

in(admit_patient,activity_class).
prop(admit_patient,input,'P',person).
prop(admit_patient,control,w,ward).
prop(admit_patient,control,phys,physician).
prop(admit_patient,control,consulting_phys,physician).
prop(admit_patient,output,pt,patient).
prop(admit_patient,actcond,a1,term(arrival,'P'§)).
prop(admit_patient,precond,'already_in?',term(not,'term(inst,'P',patient§)§)).
prop(admit_patient,precond,'room_left?',term(lt,'term(of,'cardinality',patient§),patient_max§)).
prop(admit_patient,postcond,'admitted?',term(in_hospital,'P'§)).
prop(admit_patient,part,check_id,term(check_id,'P'§)).
prop(admit_patient,part,temp,term(take_temp,'P'§)).

```

```

in(admit_child_patient,activity_class).
isa(admit_child_patient,admit_patient).
prop(admit_child_patient,input,'P',child).
prop(admit_child_patient,control,n,nurse).
prop(admit_child_patient,output,pt,child_patient).
prop(admit_child_patient,part,find_nurse,term(find_nurse,'n,w§)).
prop(admit_child_patient,part,admit,term(insert,'P',child_patient§)).

```

```

in(admit_surgical_patient,activity_class).
isa(admit_surgical_patient,admit_patient).
prop(admit_surgical_patient,output,pt,surgical_patient).
prop(admit_surgical_patient,actcond,a1,term(and,'term(arrival,'P'§),term(surgery_needed,'P'§)§)).
prop(admit_surgical_patient,part,blood_typing,perform_blood_typing).

```

```

in(admit_surgical_child_patient,activity_class).
isa(admit_surgical_child_patient,admit_child_patient).
isa(admit_surgical_child_patient,admit_surgical_patient).
prop(admit_surgical_child_patient,part,obtain_permission,term(obtain_permission,'P',term(of,'guardian','P'§)§)).

```

ANNEXE 2.9.2

Ex2.PLf

```
admit_patient(P) :-
    verify(P, person),
    not inst(P, patient),
    prop(patient, _207, cardinality, _195),
    lt(_195, patient_max),
    check_id(P),
    take_temp(P),
    in_hospital(P),
    remove(P, person),
    true.

admit_child_patient(P) :-
    verify(P, child),
    not inst(P, patient),
    prop(patient, _236, cardinality, _226),
    lt(_226, patient_max),
    find_nurse(n, w),
    insert(P, child_patient),
    check_id(P),
    take_temp(P),
    in_hospital(P),
    remove(P, child),
    true.

admit_surgical_patient(P) :-
    verify(P, person),
    not inst(P, patient),
    prop(patient, _236, cardinality, _226),
    lt(_226, patient_max),
    perform_blood_typing,
    check_id(P),
    take_temp(P),
    in_hospital(P),
    remove(P, person),
    true.

admit_surgical_child_patient(P) :-
    verify(P, child),
    not inst(P, patient),
    prop(patient, _393, cardinality, _381),
    lt(_381, patient_max),
    prop(P, _220, guardian, _210),
    obtain_permission(P, _210),
    find_nurse(n, w),
    insert(P, child_patient),
    check_id(P),
    take_temp(P),
    perform_blood_typing,
    in_hospital(P),
    remove(P, child),
    true.
```


Ex3.7.7

```

in(entre,entity_class).
isa(entre,entity).
prop(entre,argument,'INF',integer).
prop(entre,argument,'SUP',integer).
prop(entre,neccond,n1,term(number,['THIS'])).
prop(entre,neccond,n2,term(1e,['THIS','SUP'])).
prop(entre,neccond,n3,term(1e,['INF','THIS'])).
in(date,entity_class).
isa(date,entity).
prop(date,part,year,term(entre,[45,84])).
prop(date,part,mounth,term(entre,[1,12])).
prop(date,part,day,term(entre,[1,31])).
in(today,date).
prop(today,part,year,84).
prop(today,part,mounth,october).
prop(today,part,day,23).
in(person,entity_class).
isa(person,entity).
prop(person,part,nom,term(string,[20])).
prop(person,part,prenom,term(string,[20])).
prop(person,part,adresse,adresse).
prop(person,part,tel,term(integer,[9])).
prop(person,part,naiss,date).
prop(person,part,age,term(2,[person_age])).
in(person_age,activity_class).
prop(person_age,control,'THIS',person).
prop(person_age,output,'AGE',term(integer,[3])).
prop(person_age,part,a1,term(is,['AGE',term(2,[term(moins,[term(of,[year,today]),term(
in(adresse,entity_class).
isa(adresse,entity).
prop(adresse,part,rue,term(string,[30])).
prop(adresse,part,no,term(integer,[3])).
prop(adresse,part,code_postal,term(integer,[4])).
prop(adresse,part,localite,term(string,[20])).
in(integer,entity_class).
prop(integer,argument,'I',number).
prop(integer,precond,p,term(number,['THIS'])).
prop(integer,neccond,n,term(1t,['THIS',term(2,[term(exp,[10,'I'])])])).
in(string,entity_class).
prop(string,argument,'I',term(integer,[3])).
prop(string,precond,p,term(atom,['THIS'])).
prop(string,neccond,n,term(1e,[term(2,[term(strlen,['THIS'])]),'I'])).
in(number,entity_class).
prop(number,neccond,n,term(number,['THIS'])).
in(js,person).
prop(js,part,nom,smith).
prop(js,part,prenom,john).
prop(js,part,adresse,adresse_js).
prop(js,part,tel,27595925).
prop(js,part,naiss,naiss_js).
in(naiss_js,date).
prop(naiss_js,part,year,60).
prop(naiss_js,part,mounth,6).
prop(naiss_js,part,day,8).
in(adresse_js,adresse).
prop(adresse_js,part,rue,du_moulin).
prop(adresse_js,part,no,64).
prop(adresse_js,part,code_postal,1000).
prop(adresse_js,part,localite,bruxelles).

```

```

in(entre,entity_class).
in(date,entity_class).
in(today,date).
in(person,entity_class).
in(person_age,activity_class).
in(adresse,entity_class).
in(integer,entity_class).
in(string,entity_class).
in(number,entity_class).
in(js,person).
in(naiss_js,date).
in(adresse_js,adresse).
isa(entre,entity).
isa(date,entity).
isa(person,entity).
isa(adresse,entity).
person_age(THIS,AGE) :-
    prop(today,_191,year,_179),
    prop(THIS,_213,naiss,_205),
    prop(_205,_206,year,_196),
    moins(_179,_196,_159),
    AGE is _159,
    verify(AGE,integer(3)),
    true.
verify_nec(THIS,entre(INF,SUP)) :-
    verify(INF,number),
    verify(SUP,number),
    number(THIS),
    le(THIS,SUP),
    le(INF,THIS),
    true.
verify_nec(THIS,date) :-
    true.
verify_nec(THIS,person) :-
    true.
verify_nec(THIS,adresse) :-
    true.
verify_nec(THIS,integer(1)) :-
    verify(1,number),
    number(THIS),
    exp(10,I,_1E1),
    lt(THIS,_1E1),
    true.
verify_nec(THIS,string(1)) :-
    verify(1,integer(3)),
    atom(THIS),
    strlen(THIS,_153),
    le(_153,1),
    true.
verify_nec(THIS,number) :-
    number(THIS),
    true.
prop(entre,argument,'INF',integer).
prop(entre,argument,'SUP',integer).
prop(date,part,year,entre(45,84)).
prop(date,part,mounth,entre(1,12)).
prop(date,part,day,entre(1,31)).
prop(person,part,nom,string(20)).
prop(person,part,prenom,string(20)).
prop(person,part,adresse,adresse).
prop(person,part,tel,integer(9)).
prop(person,part,naiss,date).

```


ANNEXE 2.10.3

EX3.FEF (2)

```
prop(adresse,part,rue,string(30)).
prop(adresse,part,no,integer(3)).
prop(adresse,part,code_postal,integer(4)).
prop(adresse,part,localite,string(20)).
prop(integer,argument,'I',number).
prop(integer,precond,p,number('THIS')).
prop(string,argument,'I',integer(3)).
prop(string,precond,p,atom('THIS')).
prop(today,part,year,84).
prop(today,part,mounth,october).
prop(today,part,day,23).
prop(naiss_js,part,year,60).
prop(naiss_js,part,mounth,6).
prop(naiss_js,part,day,8).
prop(js,part,nom,smith).
prop(js,part,prenom,john).
prop(js,part,adresse,adresse_js).
prop(js,part,tel,27595925).
prop(js,part,naiss,naiss_js).
prop(adresse_js,part,rue,du_moulin).
prop(adresse_js,part,no,64).
prop(adresse_js,part,code_postal,1000).
prop(adresse_js,part,localite,bruxelles).
```

ANNEXE 3

LE SYSTEME MIZAR

Le langage MIZAR est un calcul des prédicats du premier ordre, multi-ensembliste, connaissant, dans le cas de MIZAR-MSE, les propriétés de l'égalité, MIZAR-MSF permettant d'introduire des fonctions.

MIZAR-MSE a été conçu pour l'enseignement des théories logique et mathématique élémentaires. Son fonctionnement est très simple : Vous disposez d'une série d'axiomes, étiquetés, portant sur des éléments d'ensembles. Ceux-ci vont constituer l'environnement de votre travail.

Lorsque vous aurez rédigé ces axiomes dans le langage MIZAR, vous pourrez alors démontrer un ou plusieurs théorèmes basés sur la théorie ainsi formée.

MIZAR vérifiera alors la correction syntaxique et logique de la démonstration, ainsi que sa complétude. Les erreurs sont soulignées et la démonstration refusée, ou bien MIZAR accepte votre raisonnement. 63 types d'erreurs ont été définis par les concepteurs.

Le noyau du système est le "checker". Son fonctionnement repose sur les principes suivants :

- 1) Accepte les tautologies.
- 2) Vérifie chaque inférence en considérant les prémisses.
- 3) Considère et traite l'égalité comme réflexive, symétrique et transitive.
- 4) Considère l'équivalence comme la conjonction de deux implications.
- 5) Refuse l'inférence dans laquelle les prémisses contiennent plus d'une proposition universellement quantifiée.
- 6) Considère que la meilleure méthode pour prouver que quelque chose existe est de trouver un exemple de ce quelque chose.

7) Utilise les règles d'inférence suivantes :

- abstraction par concrétisation
- loi de transition d'un cas général à un cas particulier.

Il serait superflu de détailler ici la syntaxe du langage. Il faut seulement savoir qu'elle est orientée-Anglais, et suffisamment simple pour être facilement compréhensible par les non initiés.

Les quelques exemples qui suivent permettent d'apprécier les capacités de MIZAR-MSE.

ENVIRON

REFLEXIVITY: FOR X BEING RATIONAL HOLDS $NGEX, X$;

TRANSITIVITY: FOR X, Y, Z BEING RATIONAL ST $NGEX, Y$ & $NGFY, Z$ _____
HOLDS $NGEX, Z$;

ANTISYMMETRY: FOR X, Y BEING RATIONAL ST $NGEX, Y$ & $NGFY, X$ _____
HOLDS $X=Y$;

CONNECTEDNESS: FOR X, Y BEING RATIONAL HOLDS $NGEX, Y$ OR $NGFY, X$;

GIVEN A, B, C, D, E BEING RATIONAL:

Z1: $A \leq D$;

Z2: $NG(A, C)$;

Z3: $NG(B, C)$;

Z4: $NG(C, D)$;

Z5: NOT $NG(A, E)$;

BEGIN

$NG(A, D)$ BY Z2, Z4, TRANSITIVITY;

THANKS OK.

ENVIRON

REFLEXIVITY: FOR X BEING RATIONAL HOLDS $NG(X,X)$:

TRANSITIVITY: FOR X,Y,Z BEING RATIONAL ST $NG(X,Y)$ & $NG(Y,Z)$
HOLDS $NG(X,Z)$:

ANTISYMMETRY: FOR X,Y BEING RATIONAL ST $NG(X,Y)$ & $NG(Y,X)$
HOLDS $X=Y$:

CONNECTEDNESS: FOR X,Y BEING RATIONAL HOLDS $NG(X,Y)$ OR $NG(Y,X)$:

GIVEN A,B,C,D,E BEING RATIONAL:

Z1: $A \neq D$:

Z2: $NG(A,C)$:

Z3: $NG(B,C)$:

Z4: $NG(C,D)$:

Z5: NOT $NG(A,E)$:

BEGIN

T1: $NG(A,D)$ BY Z2,Z4,TRANSITIVITY;

NOT $NG(D,A)$ BY Z1,T1,ANTISYMMETRY;

THANKS OK

ENVIRON

REFLEXIVITY: FOR X BEING RATIONAL HOLDS $NGEX, X1$:

TRANSITIVITY: FOR X,Y,Z BEING RATIONAL ST $NGEX, Y1$ & $NGCY, Z1$
HOLDS $NGEX, Z1$:

ANTISYMMETRY: FOR X,Y BEING RATIONAL ST $NGEX, Y1$ & $NGCY, X1$
HOLDS $X=Y$:

CONNECTEDNESS: FOR X,Y BEING RATIONAL HOLDS $NGEX, Y1$ OR $NGCY, X1$:

GIVEN A,B,C,D,E BEING RATIONAL:

Z1: $A < B$:

Z2: $NG(A, B)$:

Z3: $NG(B, C)$:

Z4: $NG(C, D)$:

Z5: NOT $NG(A, E)$:

BEFIN

T3: $NG(A, A)$ BY REFLEXIVITY:

EX X BEING RATIONAL ST $NGEX, X1$ BY T3:

THANKS OK

ENVIRON

REFLEXIVITY: FOR X BEING RATIONAL HOLDS $NG[X,X]$;

TRANSITIVITY: FOR X,Y,Z BEING RATIONAL ST $NG[X,Y]$ & $NG[Y,Z]$
HOLDS $NG[X,Z]$;

ANTISYMMETRY: FOR X,Y BEING RATIONAL ST $NG[X,Y]$ & $NG[Y,X]$
HOLDS $X=Y$;

CONNECTEDNESS: FOR X,Y BEING RATIONAL HOLDS $NG[X,Y]$ OR $NG[Y,X]$;

GIVEN A,B,C,D,E BEING RATIONAL;

Z1: $A \neq D$;

Z2: $NG[A,C]$;

Z3: $NG[B,C]$;

Z4: $NG[C,D]$;

Z5: NOT $NG[A,E]$;

BEGIN

FOR X,Y,Z BEING RATIONAL
ST NOT $NG[X,Z]$ & $NG[Y,Z]$ HOLDS NOT $NG[X,Y]$

PROOF

LET X',Y',Z' BE RATIONAL SUCH THAT

A: NOT $NG[X',Z']$ AND B: $NG[Y',Z']$;

L1: $X' \neq Y'$ BY A,B;

L2: $NG[Z',X']$ BY A,CONNECTEDNESS;

L3: $NG[Y',X']$ BY L2,B,TRANSITIVITY;

THUS NOT $NG[X',Y']$ BY L1,L3,ANTISYMMETRY

END;

THANKS OK

ENVIRON

REFLEXIVITY: FOR X BEING RATIONAL HOLDS $NG[X,X]$;

TRANSITIVITY: FOR X,Y,Z BEING RATIONAL ST $NG[X,Y]$ & $NG[Y,Z]$
HOLDS $NG[X,Z]$;

ANTISYMMETRY: FOR X,Y BEING RATIONAL ST $NG[X,Y]$ & $NG[Y,X]$
HOLDS $X=Y$;

CONNECTEDNESS: FOR X,Y BEING RATIONAL HOLDS $NG[X,Y]$ OR $NG[Y,X]$;

GIVEN A,B,C,D,E BEING RATIONAL;

Z1: $A \neq D$;

Z2: $NG[A,C]$;

Z3: $NG[B,C]$;

Z4: $NG[C,D]$;

Z5: NOT $NG[A,E]$;

BEGIN

FOR X,Y,Z BEING RATIONAL
ST NOT $NG[X,Y]$ & NOT $NG[Y,Z]$ HOLDS NOT $NG[X,Z]$

PROOF

LET X,Y,Z BE RATIONAL SUCH THAT
A: NOT $NG[X,Y]$ AND B: NOT $NG[Y,Z]$;
ASSUME C: $NG[X,Z]$;
D: $NG[Z,Y]$ BY B, CONNECTEDNESS;
E: $NG[X,Y]$ BY C,D, TRANSITIVITY;
THUS CONTRADICTION BY E,A

END;

THANKS OK

ENVIRON

LET X, Y, Z, X', Y' DENOTE RATIONAL;

REFLEXIVITY: FOR X HOLDS $NGEX, X$;

TRANSITIVITY: FOR X, Y, Z ST $NGEX, Y$ & $NGEY, Z$
HOLDS $NGEX, Z$;

ANTISYMMETRY: FOR X, Y ST $NGEX, Y$ & $NGEY, X$
HOLDS $X=Y$;

CONNECTEDNESS: FOR X, Y HOLDS $NGEX, Y$ OR $NGEY, X$;

INFINITY: FOR X HOLDS $(\exists Y \text{ ST } X < Y \text{ \& } NGEX, Y) \text{ \& }$
 $(\exists Z \text{ ST } X < Z \text{ \& } NGEZ, X)$;

DENSITY: FOR X, Y ST $X < Y$ & $NGEX, Y$
 $\exists Z \text{ ST } NGEX, Z$ & $NGEZ, Y$ & $X < Z$ & $Z < Y$;

GIVEN A, B, C, D, E BEING RATIONAL;

Z1: $A < D$;

Z2: $NGEA, C$;

Z3: $NGEB, C$;

Z4: $NGEC, D$;

Z5: NOT $NGEA, E$;

BEGIN

FOR X, Y ST $X < Y$ & $NGEX, Y$

EX X', Y' ST NOT $NGEX', X$ & NOT $NGEY', X'$ & NOT $NGEY, Y'$

PROOF

LET X, Y BE RATIONAL SUCH THAT $A: X < Y$ AND $B: NGEX, Y$;
 $\exists Z \text{ ST } NGEX, Z$ & $NGEZ, Y$ & $X < Z$ & $Z < Y$ BY A, D, DENSITY;
THEN CONSIDER X' SUCH THAT $C: NGEX, X'$ AND $D: NGEX', Y$ AND
 $E: X < X'$ AND $F: X' < Y$;

$K: \text{NOT } NGEX', Y$ BY C, F, ANTISYMMETRY;

NOT $NGEY, X'$ BY D, E, ANTISYMMETRY;

THEN $NGEX', Y$ BY CONNECTEDNESS;

THEN $\exists Z \text{ ST } NGEX', Z$ & $NGEZ, Y$ & $X' < Z$ & $Z < Y$ BY F, DENSITY;

THEN CONSIDER Y' SUCH THAT $G: NGEX', Y'$ AND $H: NGEY', Y$ AND
 $I: X' < Y'$ AND $J: Y' < Y$;

$L: \text{NOT } NGEY', X'$ BY G, I, ANTISYMMETRY;

NOT $NGEY, Y'$ BY H, J, ANTISYMMETRY;

HENCE THESIS BY K, L

END;

THANKS OK

ENVIRON

LET X,Y,Z DENOTE RATIONAL;

REFLEXIVITY: FOR X HOLDS NGEX,XJ;

TRANSITIVITY: FOR X,Y,Z ST NGEX,YJ & NGCY,ZJ
HOLDS NGCX,ZJ;

ANTISYMMETRY: FOR X,Y ST NGEX,YJ & NGCY,XJ
HOLDS X=Y;

CONNECTEDNESS: FOR X,Y HOLDS NGEX,YJ OR NGCY,XJ;

INFINITY: FOR X HOLDS (EX Y ST X<>Y & NGEX,YJ) &
(EX Z ST X<>Z & NGCZ,XJ);

DENSITY: FOR X,Y ST X<>Y & NGEX,YJ
EX Z ST NGCX,ZJ & NGCZ,YJ & X<>Z & Z<>Y;

BETWEEN: FOR X,Y,Z HOLDS BCX,Y,ZJ IFF NGEX,YJ & NGCY,ZJ;

GIVEN A,B,C,D,E BEING RATIONAL;

Z1: A<>B;

Z2: NGEA,CJ;

Z3: NGER,CJ;

Z4: NGCC,DJ;

Z5: NOT NGER,EJ;

BEGIN

FOR X,Y HOLDS X<>Y IFF
(NOT NGEX,YJ OR NOT NGCY,XJ)

PROOF

LET X,Y BE RATIONAL;

THUS X<>Y IMPLIES (NOT NGEX,YJ OR NOT NGCY,XJ)

PROOF

ASSUME A:X<>Y;

ASSUME NGEX,YJ & NGCY,XJ;

THEN X=Y BY ANTISYMMETRY;

HENCE CONTRADICTION BY A

END;

THUS (NOT NGEX,YJ OR NOT NGCY,XJ) IMPLIES X<>Y

PROOF

ASSUME B:NOT NGEX,YJ OR NOT NGCY,XJ;

ASSUME X=Y;

THEN NGEX,YJ & NGCY,XJ BY REFLEXIVITY;

HENCE CONTRADICTION BY B

END

END;

THANKS OR

Jusqu'à présent, MIZAR a toujours été utilisé pour l'enseignement des mathématiques, mais il est très probable qu'il soit, dans un avenir proche, appliqué à d'autres domaines comme l'économie, et pourquoi pas le droit ?

Dans le cas d'un jugement, on pourrait très bien considérer les règles de droit utilisées comme les axiomes de la théorie, le jugement final étant le théorème à prouver, et l'argumentation devenant la démonstration du théorème.

Un jugement étant souvent assez hétéroclite, il serait sans doute plus sage de réduire le théorème à prouver à une composition de ce jugement. Bien entendu, il serait nécessaire de formaliser les principes de droit et les jugements sous forme d'une logique du premier ordre, et d'ajouter des axiomes interdisant, par exemple, l'occurrence simultanée de conséquences contradictoires.